

Aplicação Móvel para Melhorar a Mobilidade na Condução

PROJETO DE MESTRADO

Ricardo Filipe Henriques Faria

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

março | 2020

Aplicação Móvel para Melhorar a Mobilidade na Condução

PROJETO DE MESTRADO

Ricardo Filipe Henriques Faria

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTAÇÃO

Karolina Baras

CO-ORIENTAÇÃO

Lina Maria Pestana Leão Brito



UNIVERSIDADE da MADEIRA

APLICAÇÃO MÓVEL PARA MELHORAR A MOBILIDADE NA CONDUÇÃO

Ricardo Filipe Henriques Faria
Mestrado em Engenharia Informática

ORIENTADORA: KAROLINA BARAS

COORIENTADORA: LINA BRITO

FUNCHAL, MARÇO 2020



Aplicação Móvel para Melhorar a Mobilidade na Condução

Ricardo Filipe Henriques Faria

Constituição do júri de provas públicas:

Eduardo Miguel Dias Marques, (Professor Auxiliar da Universidade da Madeira), Presidente

Filipe Magno de Gouveia Quintal, (Professor Auxiliar Convidado da Universidade da Madeira), Vogal

Lina Maria Pestana Leão de Brito, (Professora Auxiliar da Universidade da Madeira), Vogal

Março 2020

Funchal – Portugal

Resumo

A ilha da Madeira, similarmente a outras regiões europeias, tem sofrido cada vez mais com o aumento da densidade do tráfego automóvel. Nas grandes zonas urbanas, a aglomeração de pessoas e automóveis tem criado situações de trânsito caótico, essencialmente devido ao planeamento urbano que não conseguiu acompanhar as exigências de mobilidade que as populações necessitam. Estes problemas relacionados com a mobilidade urbana levaram a que os governantes tivessem de adotar novas medidas para minimizar o impacto do aumento de veículos a circular nas estradas regionais.

Embora se tenham verificado muitas melhorias, ainda assim, os condutores frequentemente experienciam situações inesperadas enquanto conduzem, tais como estradas encerradas devido a obras, eventos desportivos ou religiosos que ocorrem nas cidades, o que os força a decidir rapidamente quais as rotas alternativas para conseguirem chegar aos seus destinos a tempo.

A solução a ser apresentada tem como principal objetivo melhorar a mobilidade dentro e fora das cidades, ajudando os cidadãos a otimizar a gestão do seu tempo, reduzindo assim o *stress* diário. Para isso, a plataforma irá centralizar todas as informações existentes sobre as condições de tráfego, tendo também como foco todos os eventos que ocorram e que obriguem assim ao encerramento de alguma via. Por outro lado, serão apresentadas todas as estradas que tenham uma inclinação acentuada ou uma largura que torne difícil a passagem de veículos. Deste modo, poderão ser apresentados, de forma prévia, alguns dados que permitirão aos condutores evitar determinadas vias que lhes causariam algum tipo de *stress*.

A plataforma desenvolvida recolhe e filtra estes dados em *background*, possibilitando, através de uma aplicação móvel, o acesso aos mesmos de uma forma detalhada e de fácil visualização. Desta forma foram preenchidas lacunas atualmente existentes noutras plataformas, permitindo mostrar aos utilizadores estas informações em tempo real, ou antecipadamente, quando possível, através de notificações.

Palavras-Chave: Aplicações Móveis, Vias Encerradas, Informações de Tráfego, Mobilidade.

Abstract

Madeira island, identically to other European regions, has been suffering more and more with the traffic congestion increase over time. In large urban areas, the agglomeration of people and vehicles has created chaotic traffic situations, essentially due to urban planning that was unable to fulfill the populations mobility requirements. These problems related to urban mobility have led governments to adopt new measures to minimize the impact of vehicles in circulation on regional roads.

Although many improvements have been made, drivers often experience unexpected situations as they drive, such as closed roads due to construction, sporting or religious events occurring in cities, forcing them to quickly decide alternate routes to reach their destinations in time.

The presented solution aims to improve mobility inside and outside of the cities, helping citizens to optimize their time management, thus reducing daily stress. To achieve this, the platform will centralize all the existing information about traffic conditions, also focusing on all events that occur and that require some road closure. On the other hand, all steep or narrow roads will be displayed in the application. Thereby, this data will be presented in advance, allowing drivers to avoid roads that could cause them stress.

The developed platform collects and filters this data in the background, enabling, through a mobile application, access to it in a way that is detailed and easy-to-see. It also allows showing this information in real-time, or in advance, when possible, by sending notifications.

Keywords: Mobile Applications, Closed Roads, Traffic Information, Mobility.

Agradecimentos

Em primeiro lugar, gostaria de agradecer à minha orientadora, Professora Karolina Baras, à minha coorientadora, Professora Lina Brito, por todo o apoio, orientações prestadas, pela disponibilidade, paciência e pelos conhecimentos que me transmitiram ao longo do desenvolvimento deste projeto de mestrado.

Quero agradecer à Câmara Municipal do Funchal por disponibilizar os dados relacionados com a largura das estradas deste mesmo concelho.

Quero também agradecer a toda a minha família que me acompanhou durante toda a formação académica, tendo-me transmitido confiança e motivação para poder finalizar mais este percurso na minha vida.

Por último, mas não menos importante, quero agradecer a todos os colegas e amigos pelo apoio e confiança que depositaram em mim e nas minhas capacidades ao longo do meu percurso académico.

A todos, um muito obrigado!

Índice

1. Introdução	1
1.1 Contextualização Demográfica	2
1.2 Contextualização Orográfica	5
1.3 Motivação	7
1.4 Objetivos	10
1.5 Estrutura do Documento	11
2. Estado da Arte	13
2.1 Redes Sociais	13
2.2 Sensores	17
2.3 Conclusões	19
3. Arquitetura de Software	22
3.1 Interação entre os Componentes	25
3.1.1 Tratamento dos Dados obtidos a partir do Serviço Regional de Proteção Civil da Madeira	26
3.1.2 Normalização dos Dados na Página de Administração	28
3.1.3 Envio dos Dados para a Aplicação	29
3.2 Atributos e Cenários de Qualidade	30
4. Desenvolvimento da Aplicação	34
4.1 Plataforma de Administração	34
4.2 DNS	37
4.3 HTTPS	38
4.4 Servidor	42
4.4.1 Módulo Civil Protection	43
4.4.2 Módulo livesql	46
4.4.3. Módulo nodemailer (nodemailer <i>package</i>)	49
4.4.4. Módulo socket_io (socket.io <i>package</i>)	52
4.4.5. Módulo https (https <i>package</i>)	53
4.5 Base de Dados	54
4.5.1 Base de Dados closed_roads	55
4.5.2 Base de Dados users	58
4.5.3 Base de Dados roads_data	58
4.6 Plataforma Android	60
4.6.1 Módulo Activity Recognition	61
4.6.2 Módulo Closed Roads	62

4.6.3	Módulo Elevations.....	62
4.6.4	Módulo WiFi.....	64
4.6.5	Módulo Roads Width.....	65
4.6.6	Módulo Ocorrências.....	65
4.6.7	Módulos Routes e Instructions.....	66
4.6.8	Interface de Utilizador.....	67
4.7	Conclusões.....	73
5.	Testes e Resultados.....	76
5.1	Tempos de Acesso (APIs) no Servidor.....	76
5.1.1	Proteção Civil.....	76
5.1.2	Overpass API (Obtenção do nome da via encerrada).....	77
5.1.3	Overpass API (Obtenção dos nós de cada via).....	78
5.1.4	MapQuest API (Obtenção dos Link Ids).....	79
5.2	Atributos e Cenários de Qualidade.....	80
5.2.1.	Disponibilidade.....	81
5.2.2.	Segurança.....	81
5.2.3.	Desempenho.....	82
5.2.4.	Modificabilidade.....	82
5.2.5.	Interoperabilidade.....	82
5.3	Questionários.....	83
5.3.1.	Discussão do resultado dos questionários.....	83
5.4	Limitações da Aplicação.....	85
6.	Conclusões.....	88
6.1	Objetivos Atingidos.....	88
6.2	Trabalho Futuro.....	89
	Referências.....	92
	Anexos.....	96

Lista de Tabelas

Tabela 1: População residente na RAM e respetivos municípios entre 2013 e 2017 [3].	2
Tabela 2: Número médio de hab./km² a nível nacional entre 2001 e 2017 [3].	3
Tabela 3: Principais características físicas da Ilha da Madeira [6].	5
Tabela 4: Cenários de qualidade ordenados por prioridade.	31
Tabela 5: Atribuição de notas utilizando o SSL Labs.	41

Lista de Figuras

Figura 1: Número médio de hab./km ² a nível nacional (zonas) entre 2001 e 2017 [3].	4
Figura 2: Mapas topográficos da Ilha da Madeira e da cidade do Funchal.	6
Figura 3: Aplicação Google Traffic na Ilha da Madeira.	8
Figura 4: Aplicação Waze na Ilha da Madeira.	9
Figura 5: Coordenadas necessárias para a criação de uma bounding box [9].	14
Figura 6: Exemplo de um post na página da polícia de trânsito de Calcutá indicando que ocorrerá uma procissão religiosa que irá obrigar ao encerramento de algumas ruas [14].	17
Figura 7: Arquitetura da Plataforma.	23
Figura 8: Página de Login.	35
Figura 9: Página Principal.	36
Figura 10: Redirecionamento dos requests.	39
Figura 11: Análise da configuração do servidor web SSL utilizando o SSL Labs.	40
Figura 12: Análise da configuração do servidor web SSL utilizando o SSL Checker.	42
Figura 13: Esquema do funcionamento dos binlogs nas bases de dados MySQL.	48
Figura 14: Formulário de ativação de aplicações menos seguras.	50
Figura 15: Email informativo da existência de um novo registo na tabela temp_civil_protection.	50
Figura 16: Exemplo de uma notificação de erro.	51
Figura 17: Tabela android_device.	55
Figura 18: Diagrama Entidade-Relação entre as quatro tabelas onde são guardados os dados obtidos a partir do Serviço Regional de Proteção Civil da Madeira.	57
Figura 19: Tabela users_login.	58
Figura 20: Diagrama Entidade - Relação entre as três tabelas onde são guardadas as informações relacionadas com a largura das estradas.	60
Figura 21: Exemplo da obtenção da inclinação nos troços das vias.	63
Figura 22: Exemplo da apresentação das manobras a realizar.	66
Figura 23: Exemplo de estrada encerrada.	67
Figura 24: Exemplo de uma estrada condicionada.	68
Figura 25: Exemplo de uma via com um elevado grau de inclinação.	69
Figura 26: Exemplo de uma estrada estreita.	69
Figura 27: Vista com os dados dos posts do "Ocorrências Madeira".	70
Figura 28: Exemplo da representação de um acidente partilhado no "Ocorrências Madeira".	71
Figura 29: Exemplo da apresentação dos locais mais frequentados pelos utilizadores (casa e trabalho).	72

Figura 30: Exemplo da apresentação das manobras a seguir pelo utilizador no mapa.....	72
Figura 31: Gráfico com os tempos de acesso ao website do Serviço Regional de Proteção Civil da Madeira.....	77
Figura 32: Gráfico com os tempos de acesso à Overpass API (Obtenção do nome da via encerrada).....	78
Figura 33: Gráfico com os tempos de acesso à Overpass API (Obtenção dos nós de cada via).	79
Figura 34: Gráfico com os tempos de acesso ao MapQuest API (Obtenção dos Link Ids).79	

Lista de Acrónimos

ACME – *Automated Certificate Management Environment*

API – *Application Programming Interface*

CA – *Certificate Authority*

CPU – *Central Processing Unit*

CSR – *Certificate Signing Request*

CSS – *Cascading Style Sheets*

DBMS – *Database Management System*

DML – *Data Manipulation Language*

DNS – *Domain Name System*

DOM – *Document Object Model*

GPS – *Global Positioning System*

GUI – *Graphical User Interface*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

HTTPS – *HyperText Transfer Protocol Secure*

IP – *Internet Protocol*

ISRG – *Internet Security Research Group*

JSON – *JavaScript Object Notation*

KML – *Keyhole Markup Language*

MB – *Megabytes*

MD2 – *Message-Digest Algorithm 2*

MD5 – *Message-Digest Algorithm 5*

OSM – *OpenStreetMap*

PDF – *Portable Document Format*

PHP – *PHP: Hypertext Preprocessor*

QL – *Query Language*

RAM – *Região Autónoma da Madeira*

REST – *Representational State Transfer*

SMTP – *Simple Mail Transfer Protocol*

SQL – *Structured Query Language*

SSL – *Secure Socket Layer*

TLS – *Transport Layer Security*

UML – *Unified Modeling Language*

URL – *Uniform Resource Locator*

XAMPP – *Cross-Platform, Apache, MariaDB, PHP, and Perl*

XML – *eXtensible Markup Language*

1. Introdução

A ilha da Madeira apresenta diversos desafios a nível da sua estrutura rodoviária. Dentro das próprias cidades ou em zonas mais a norte da ilha, existem vias que apresentam diferentes adversidades e que dificultam a passagem a condutores mais inexperientes. Quer sejam estradas estreitas ou inclinadas, ou até mesmo o encerramento de determinadas vias, estas contrariedades elevam o nível de *stress* dos condutores, tanto devido à escassez de alternativas ou até mesmo pelas manobras necessárias para sair dessas mesmas vias.

Por estas e pelas razões a seguir apresentadas foi decidido o desenvolvimento de uma aplicação Android que pudesse ajudar a colmatar muitos destes problemas. Neste sentido, a solução a ser apresentada tem como principal objetivo melhorar a mobilidade dentro e fora das cidades, ajudando os cidadãos a otimizar a gestão do seu tempo, reduzindo assim o *stress* diário. Para isso, a plataforma irá centralizar todas as informações existentes sobre as condições de tráfego, tendo também como foco todos os eventos que ocorram e que obriguem assim ao encerramento de alguma via. Por outro lado, serão apresentadas todas as estradas que tenham uma inclinação acentuada ou uma largura que torne difícil a passagem de veículos. Deste modo é pretendida a diminuição do *stress* diário que muitos dos condutores sofrem para chegar aos seus destinos. Muitas destas situações levam a atrasos, levando a que muitos dos condutores se exaltem ou sofram de algum tipo de *stress* apenas por terem de cruzar com este tipo de situações.

A antiga rede viária madeirense era maioritariamente constituída por estradas estreitas, sinuosas e algo perigosas. A sua grande maioria encontrava-se junto às escarpas, onde normalmente existia um grande risco de derrocadas, ou então era composta por caminhos florestais ou agrícolas. A construção de novas vias, túneis, pontes e viadutos veio combater os obstáculos geológicos de uma ilha que, para além de ser uma região insular e ultraperiférica, possui uma orografia extremamente acidentada. Devido a este desenvolvimento que veio modernizar a rede viária já existente, é possível, hoje em dia, deslocar-se rapidamente entre as várias localidades da ilha, evitando deste modo que os condutores tenham de passar por estradas situadas nas vertentes das falésias, oferecendo assim uma maior mobilidade de bens e pessoas, bem como uma maior segurança e conforto [1].

Dois terços da superfície da ilha da Madeira encontram-se cobertos por vegetação, existindo uma maior densidade populacional junto às regiões costeiras. A ocupação do território na ilha da Madeira permite verificar a existência de uma maior densidade populacional no litoral sul, de forma especialmente intensa na mancha contínua que se localiza entre os concelhos do Funchal, Câmara de Lobos e Santa Cruz. No entanto, este aumento demográfico nas regiões mais a sul apenas se verificou nas últimas décadas, contrastando com o crescimento populacional, que até então se verificava especialmente nos concelhos rurais. Esta situação deveu-se ao progressivo abandono das atividades ligadas ao setor primário, que foram substituídas pelo setor terciário, nomeadamente a hotelaria. Verificou-se então um rápido crescimento urbano, particularmente no concelho do Funchal, onde foram criados novos eixos rodoviários, que permitiram o alargamento da cidade até às zonas montanhosas circundantes [2].

1.1 Contextualização Demográfica

A ilha da Madeira é a principal ilha do arquipélago da Madeira (Região Autónoma da Madeira), arquipélago que também é composto pela ilha do Porto Santo, pelas ilhas Desertas e pelas ilhas Selvagens. Esta situa-se no Oceano Atlântico, a sudoeste da costa portuguesa e tem como capital a cidade do Funchal.

A Região Autónoma da Madeira (RAM) tinha, segundo dados da Direção Regional de Estatística da Madeira de 2017 [3], uma população de aproximadamente 254 368 pessoas, das quais 104 442 residiam na sua capital, a cidade do Funchal, o que correspondia a cerca de 41% da população total (Tabela 1).

	R. A. Madeira	Calheta	Câmara de Lobos	Funchal	Machico	Ponta do Sol	Porto Moniz	Ribeira Brava	Santa Cruz	Santana	São Vicente	Porto Santo
	População total (Nº)											
2013	261 313	11 295	34 798	108 053	21 147	8 787	2 530	12 889	43 827	7 272	5 416	5 299
2014	258 686	11 170	34 482	106 721	20 874	8 694	2 469	12 720	43 873	7 131	5 301	5 251
2015	256 424	11 052	34 246	105 562	20 654	8 619	2 417	12 555	43 925	6 992	5 216	5 186
2016	254 876	10 946	34 047	104 813	20 453	8 557	2 390	12 446	44 026	6 876	5 160	5 162
2017	254 368	10 901	33 847	104 442	20 272	8 559	2 370	12 428	44 417	6 808	5 151	5 173

Tabela 1: População residente na RAM e respetivos municípios entre 2013 e 2017 [3].

Na Tabela 2 são apresentados os dados relativos ao número de habitantes por quilómetro quadrado a nível nacional, tendo como foco os dados relacionados com a RAM.

Territórios	N.º médio de indivíduos por Km²	
	2001	2017
Portugal	112,5	111,7
— Continente	110,9	110,0
+ Norte	173,2	168,2
+ Centro	83,3	79,4
+ Área Metropolitana de Lisboa	899,6	937,7
+ Alentejo	24,6	22,6
+ Algarve	79,5	88,2
+ Região Autónoma dos Açores	104,2	105,3
— Região Autónoma da Madeira	313,6	317,7
— Região Autónoma da Madeira	313,6	317,7
— Ilha da Madeira	324,4	328,9
Calheta [R.A.M.]	108,1	98,0
Câmara de Lobos	669,4	650,7
Funchal	1.377,3	1.373,8
Machico	318,4	298,4
Ponta do Sol	174,3	185,2
Porto Moniz	35,3	28,7
Ribeira Brava	193,0	190,0
Santa Cruz	448,9	542,6
Santana	91,2	71,7
São Vicente	76,5	65,4
— Ilha de Porto Santo	112,7	120,1
Porto Santo	112,7	120,1

Tabela 2: Número médio de hab./km² a nível nacional entre 2001 e 2017 [3].

Com uma densidade populacional de cerca de 317,7 hab./km², a RAM encontra-se acima da média nacional de 111,7 hab./km², apenas sendo ultrapassada pela Área Metropolitana de Lisboa (cerca de 937,7 hab./km²). No entanto, na RAM existem municípios com uma menor densidade populacional, tais como o Porto Moniz (cerca de 28,7 hab./km²), São Vicente (cerca de 65,4 hab./km²) e Santana (cerca de 71,7 hab./km²), que contrastam com outros municípios que apresentam uma maior densidade populacional, sendo estes os municípios do Funchal (cerca de 1373,8 hab./km²), Câmara de Lobos (cerca de 650,7 hab./km²) e Santa Cruz (cerca de 542,6 hab./km²).

Na Figura 1 é apresentada a densidade populacional da cidade do Funchal comparativamente com outras zonas do país. Em 2017, a cidade do Funchal encontrava-se em 16º lugar nas áreas com uma maior densidade populacional, sendo que as zonas com maior densidade

populacional em Portugal eram as zonas da Amadora (cerca de 7529,7 hab./km²), Odivelas (cerca de 5913,9 hab./km²), Porto (cerca de 5175,1 hab./km²) e Lisboa (cerca de 5052,7 hab./km²).

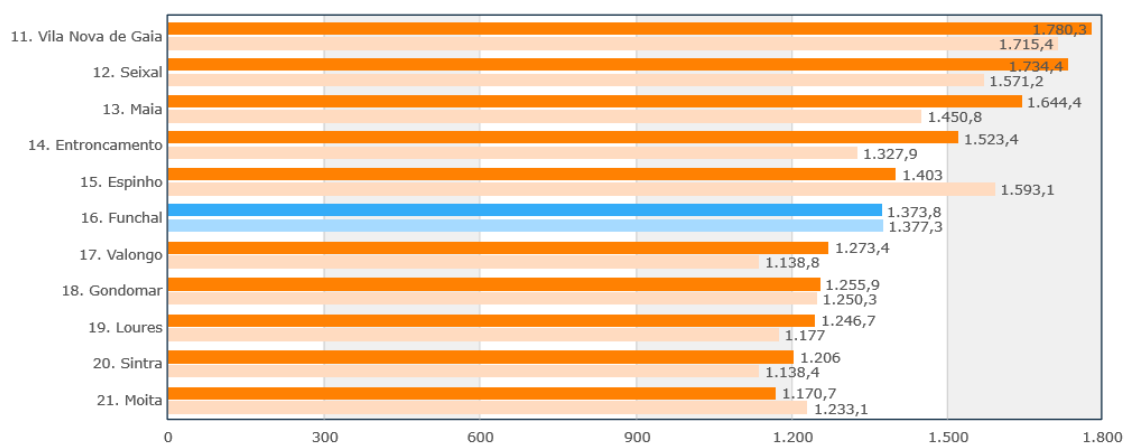


Figura 1: Número médio de hab./km² a nível nacional (zonas) entre 2001 e 2017 [3].

Como constatado anteriormente, apesar de a densidade populacional ser superior à média do país e até mesmo da União Europeia (cerca de 117,7 hab./km²)¹, cerca de 75% da população da ilha da Madeira habitava em apenas 35% do território, maioritariamente na costa sul [4].

Algumas das contribuições para este aumento demográfico que levaram a uma sobrecarga da rede viária e consequentemente dos parques de estacionamento públicos e privados, especialmente na cidade do Funchal foram:

- O êxodo rural em direção à cidade do Funchal e consequentemente aos municípios circundantes (Câmara de Lobos e Santa Cruz), que levaram a uma desertificação dos municípios mais a norte da ilha. Isto deveu-se em parte, à progressiva perda de importância da agricultura como atividade económica e a consequente diminuição da população ligada a este setor;
- O crescimento da cidade encontra-se fortemente condicionado por obstáculos naturais, que são em parte responsáveis pela forma e estrutura da ocupação do solo;
- A cidade do Funchal atrai um considerável número de população flutuante [5], uma vez que é onde se encontram os principais serviços públicos, bem como muitos dos postos de trabalho dos madeirenses;

¹ <https://ec.europa.eu/eurostat/databrowser/view/tps00003/default/table>

- A indústria hoteleira, uma das principais e mais lucrativas atividades económicas da RAM, concentra-se na cidade do Funchal, absorvendo cerca de 13 000 das 15 000 camas existentes.

Todas estas situações, juntamente com o aumento das taxas de turismo levou a um aumento das necessidades de deslocação e consequentemente a uma perda de mobilidade no centro da cidade. É reconhecido que a rede viária existente não satisfaz as necessidades atuais, provocando situações de congestionamento, verificando-se ainda a necessidade de reformular a política de estacionamento atualmente praticada, por forma a evitar indesejáveis estrangulamentos ou sobredimensionamentos [5].

1.2 Contextualização Orográfica

A ilha da Madeira ocupa uma área de cerca de 742 km², apresentando uma forma alongada na direção Este-Oeste, com um comprimento máximo de 58 km e uma largura de 23 km (Tabela 3).

Altitude média	646 m
Pico mais alto	Pico Ruivo (1862 m)
Comprimento máximo	58 km
Largura máxima	23 km
Declive médio	56%
Perímetro	177,3 km
Área	742 km ²

Tabela 3: Principais características físicas da Ilha da Madeira [6].

As encostas norte e sul da ilha da Madeira possuem características que as diferenciam. Enquanto que a encosta norte da ilha é caracterizada por um relevo vigoroso com arribas altas e escarpadas e com pouca ocupação humana, a encosta sul da ilha possui um relevo mais brando e é onde se encontra grande parte da população [7].

A Madeira é, portanto, uma ilha montanhosa (Figura 2), dominada por uma paisagem vulcânica, onde predominam os vales profundos e as encostas escarpadas. Apenas 16% da ilha tem um declive inferior a 30°, apresentando um terreno muito acidentado [8]. Uma das estradas mais antigas da ilha da Madeira, é também uma das mais peculiares e perigosas². A estrada Regional 101 (ER101), é por outro lado uma das

² <http://www.dangerousroads.org/europe/portugal/115-er101-antigua-portugal.html>

mais bonitas da ilha da Madeira, pois serpenteia a linha costeira da ilha, possuindo por isso muitas curvas, passagens estreitas e é relativamente íngreme em quase toda a sua extensão. Devido às belas paisagens que esta e também outras estradas antigas proporcionam, são muitas vezes escolhidas pelos turistas para melhor poderem conhecer as paisagens da região.

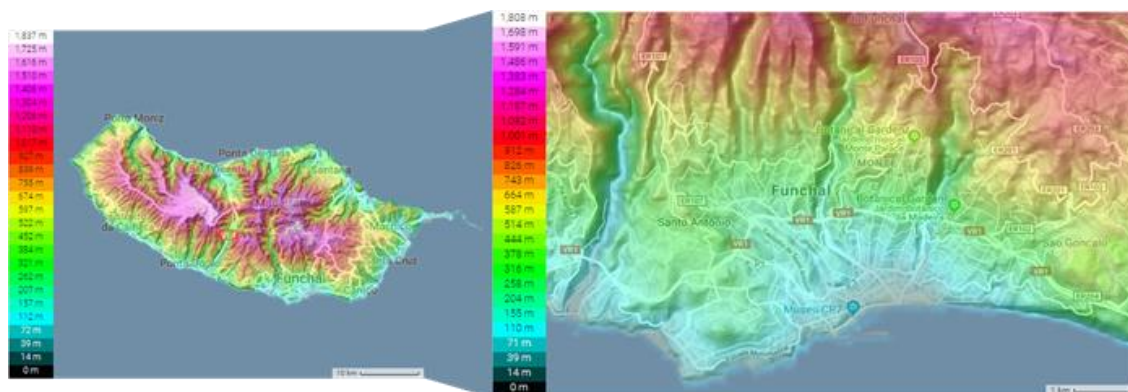


Figura 2: Mapas topográficos da Ilha da Madeira e da cidade do Funchal³.

Durante as últimas décadas, têm existido grandes obras de construção tendo em vista a ampliação da rede viária regional, visando também o melhoramento da acessibilidade às zonas mais remotas da ilha. A rede de estradas pavimentadas e a construção de vários túneis melhoraram consideravelmente os acessos e aceleraram o desenvolvimento um pouco por toda a região. Também no Funchal, a capital da RAM, foram verificadas grandes melhorias tanto na quantidade como na qualidade dos acessos ao centro da cidade. No entanto, estas melhorias não conseguiram ainda colmatar um dos grandes problemas da cidade do Funchal, que como outras grandes cidades europeias, ainda luta contra o problema dos congestionamentos de trânsito. As causas que levam a grandes engarrafamentos na cidade do Funchal são muitas. A sua orografia com estradas estreitas e íngremes, que não deixam o trânsito fluir. O próprio planeamento do território, que não foi inicialmente projetado para a quantidade de trânsito atualmente existente dentro da cidade. No centro da cidade existem escolas e outras infraestruturas que fazem com que o trânsito fique caótico nas horas de ponta. No caso das escolas, o que faz com que exista um incremento do número de veículos a circular são as situações em que os pais deixam ou buscam os filhos dos seus respetivos estabelecimentos de ensino. Outro dos problemas tem a ver com a quantidade de madeirenses que trabalham no Funchal, o que

³ <https://en-in.topographic-map.com/maps/pz24/Funchal/>

aumenta o tráfego automóvel, principalmente nas principais artérias de acesso ao centro da cidade.

1.3 Motivação

Com o crescimento do tráfego automóvel é desejável ter um mecanismo pelo qual as pessoas possam conhecer, em tempo real, o estado do trânsito. Como resultado deste problema, a monitorização do trânsito ganhou uma atenção significativa nos últimos tempos. No entanto, muitas das soluções existentes não conseguem solucionar os problemas de mobilidade urbana atualmente existentes na ilha da Madeira, tendo em conta as suas dificuldades orográficas e demográficas.

A mobilidade urbana, relacionada com as condições de tráfego, é um dos fatores-chave para o desenvolvimento das cidades inteligentes, sendo um dos maiores desafios do século XXI. O congestionamento de trânsito, as estradas interrompidas devido a obras, os eventos desportivos ou as celebrações festivas são algumas das situações que influenciam a forma como os condutores circulam pela cidade. Estas situações inesperadas exigem que os condutores encontrem soluções rápidas para alcançar os seus destinos atempadamente.

Ao utilizar um sistema de navegação GPS, estes problemas podem ser minimizados, pois estes sugerem soluções alternativas aos condutores. No entanto, a maioria das pessoas já experienciou situações em que as aplicações não os informaram acerca das mudanças mais recentes no trânsito, como por exemplo em relação a mudanças no sentido do tráfego. Os sistemas de navegação GPS criam outras situações stressantes, tais como guiar os condutores para ruas demasiado estreitas, onde estes acabam por sentir dificuldades em manobrar o veículo ou levam-nos para o meio da floresta. Além disso, os sistemas de navegação atuais não têm em consideração a inclinação das estradas, o que também pode ser um problema, especialmente se os condutores forem guiados para estradas demasiado íngremes. Portanto, torna-se importante desenvolver soluções que antecipem estas situações e que permitam alertar os condutores.

Outra das motivações para este projeto foi a falta de informações acerca das condições de trânsito na cidade do Funchal. O Google Traffic ⁴ (Figura 3) exibe as condições de trânsito em tempo real, tendo em consideração o fluxo de utilizadores num determinado trecho de estrada, ao aproveitar os

⁴ <https://www.google.com/maps/>

smartphones dos utilizadores para calcular a velocidade dos veículos ao longo da estrada. No entanto, para determinadas áreas, este apenas tem informações acerca das estradas principais. E quando o projeto foi iniciado, as informações ainda eram mais escassas uma vez que este apenas possuía informações acerca de alguns troços da Via Rápida 1 (VR1). Outros problemas encontrados foram a inexistência de informação sobre as estradas fechadas ou condicionadas, sobre as condições das vias (p. ex. buracos na via) e outras informações adicionais acerca das estradas (p. ex. largura das estradas, tipo de calçada...).

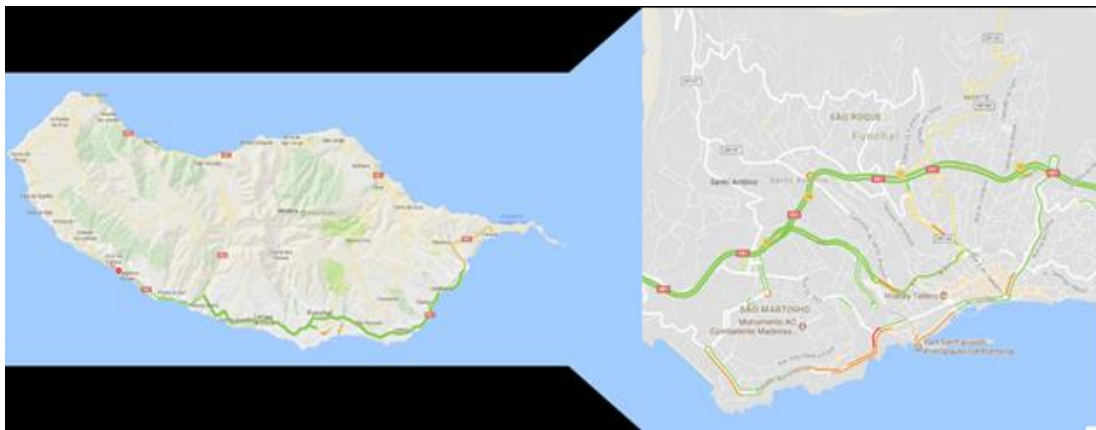


Figura 3: Aplicação Google Traffic na Ilha da Madeira.

Outra possível solução seria a aplicação Waze ⁵ (Figura 4) que também permite a partilha de informações de trânsito em tempo real, sendo atualizada pelos próprios utilizadores (permite adicionar vários eventos relacionados com o trânsito, tais como acidentes ou estradas fechadas devido a obras). Mas, para a ilha da Madeira, existe uma falta de utilizadores e de contribuições. Além disso, as informações tornam-se desatualizadas, por não haver contribuições, e também devido a não existir ninguém que controle essas mesmas contribuições. Por outro lado, também existem informações imprecisas no mapa, dando como exemplo a apresentação de nomes de estradas errados, que poderão levar a que os utilizadores se percam, caso estejam à procura de uma rua específica.

⁵ <https://www.waze.com/pt-PT/>



Figura 4: Aplicação Waze na Ilha da Madeira.

Outros problemas encontrados nas aplicações acima mencionadas, mas também em outros sistemas de navegação GPS são:

- Guiar os condutores para estradas íngremes. Conduzir na Madeira pode ser um desafio tanto para turistas como para residentes devido às suas sinuosas estradas montanhosas. Estas podem ser consideradas um labirinto contínuo que levam os condutores para colinas com um elevado grau de inclinação;
- Guiar os condutores para estradas estreitas. Nas zonas antigas, a condução pode-se tornar ainda mais stressante. As ruas são estreitas e existem muitos cruzamentos onde a própria sinalização faz com que seja difícil de entender quem tem prioridade;
- Inexperiência dos turistas em conduzir nas estradas madeirenses. Os turistas, ao não estarem familiarizados com a ilha ou com alguma parte da ilha, acabam muitas vezes por necessitar de ajuda para retirarem os seus carros de aluguer de algumas estradas;
- A inexistência de informação (aplicações atualmente existentes) relacionada com as estradas fechadas e acidentes na via.

Para além disso, numa ilha onde frequentemente existem quebradas, aluviões e outras situações que levam ao encerramento de estradas, é ainda mais importante alertar os condutores antes de estes seguirem em direção aos seus destinos.

Existe ainda outro problema, que não está relacionado com o GPS em si, mas com os regulamentos existentes nos diversos países de onde vêm os nossos turistas. Uma vez que os britânicos conduzem pelo lado esquerdo, e sendo estes uma grande percentagem dos turistas que nos visitam todos os anos, é importante ter em conta este fator, pois na Madeira a regulação exige que conduzamos pelo lado direito. Certamente, este problema não pode ser resolvido pela aplicação em si, mas serve para melhor entender alguns dos

desafios que os turistas e residentes enfrentam ao conduzir nas estradas madeirenses. Por vezes, apenas ter de conduzir pode ser uma situação stressante e, sem conhecer de antemão as estradas, tendo que se adaptar para dirigir no lado contrário ao que estamos habituados, acaba por colocar ainda mais *stress* sobre os condutores.

1.4 Objetivos

Os principais objetivos do projeto consistem no desenvolvimento de uma aplicação para Android que permita visualizar a informação acerca das estradas fechadas ou condicionadas na ilha da Madeira, obter notificações acerca das mesmas ainda antes de o utilizador iniciar a rota para o seu destino e também informar acerca das rotas a tomar caso alguma estrada se encontre encerrada ou condicionada. Este projeto tem em vista o melhoramento da qualidade de vida nas cidades, focando numa maior mobilidade e na redução do *stress* diário que afeta todos os condutores, ao fornecer um serviço para uma gestão de tempo mais eficiente.

O objetivo de reduzir o *stress* diário prende-se na ideia de que ao fornecer diferentes alternativas e ao informar os condutores antecipadamente acerca de eventos que influenciam as rotas para os seus destinos, será possível proporcionar uma melhoria significativa em termos de saúde e segurança para aqueles que todos os dias circulam nas estradas madeirenses.

Outro dos objetivos pretendidos é o desenvolvimento de uma plataforma de serviços *web* que forneça suporte à aplicação, de modo a recolher informações acerca das estradas fechadas ou condicionadas, de modo a facilitar a atualização da informação, a execução de processos e a leitura de dados que pudessem exigir elevados recursos de *hardware* por parte dos dispositivos móveis, sobrecarregando os mesmos, em termos de memória e processamento. Desta forma será também possível a centralização de toda a informação, possibilitando assim um melhor controlo da qualidade e quantidade dos dados a serem enviados e apresentados pela aplicação.

Por fim, é importante salientar o objetivo de preencher algumas lacunas existentes em outras soluções atualmente existentes, tendo em foco os utilizadores que se encontrem na ilha da Madeira. Deste modo serão tidas em conta todas as peculiaridades orográficas e demográficas da ilha, bem como todos os problemas que advêm da sua insularidade.

1.5 Estrutura do Documento

Este capítulo pretende contextualizar os problemas atualmente existentes e as respetivas motivações que levaram ao desenvolvimento deste projeto, descrevendo também os principais objetivos que funcionaram como linhas de orientação para a solução a ser apresentada.

O segundo capítulo apresenta um resumo acerca das conclusões obtidas a partir de alguns estudos e plataformas que visam resolver problemas relacionados com o tráfego automóvel e que têm como objetivo melhorar a vida dos cidadãos ao fornecer diversas informações acerca de eventos relacionados com o trânsito. Serão também salientadas algumas características em que a plataforma apresentada seguiu as ideias fornecidas pelas soluções existentes.

O terceiro capítulo esquematiza os diversos componentes que compõem a arquitetura da plataforma desenvolvida, destacando para além disso os requisitos funcionais do sistema. Por fim, é feita uma análise detalhada abordando as tecnologias disponíveis utilizadas para a implementação do sistema em questão.

O quarto capítulo descreve o desenvolvimento dos vários componentes da plataforma, pormenorizando os vários intervenientes, as ferramentas utilizadas e a forma como foram implementadas.

O quinto capítulo descreve os testes efetuados sobre a plataforma, sendo especificados os resultados obtidos e as conclusões desses mesmos resultados. São também apresentados alguns testes efetuados abordando os *timings* de acesso às APIs utilizadas e também os *timings* de acesso da respetiva aplicação ao servidor.

Por fim, no sexto capítulo são apresentadas as conclusões que foram obtidas a partir do desenvolvimento do projeto, assim como alguns aspetos que poderão ser desenvolvidos em futuras iterações.

2. Estado da Arte

Neste capítulo será abordada a literatura atualmente existente, explorando as diferentes técnicas e métodos utilizados, bem como as várias ferramentas que serviram como base para as soluções apresentadas. Esta informação será então recolhida e analisada, tendo como principal objetivo a obtenção de um conhecimento sólido acerca das plataformas e estudos já existentes. Ao identificar o que já foi realizado, será possível a utilização de técnicas semelhantes, bem como o desenvolvimento de melhorias baseadas nas lacunas encontradas. Basicamente, neste capítulo, serão descritos alguns dos estudos que se baseiem na obtenção de dados sobre eventos que influenciem o tráfego automóvel, quer sejam dados provenientes de serviços *web*, quer sejam dados obtidos a partir de sensores e, que no fundo, permitam facilitar a vida dos condutores ao longo dos seus trajetos.

Muitos destes estudos baseiam-se na obtenção de dados em bruto, apresentando técnicas e ferramentas que possibilitem a respetiva filtragem e análise destes dados, antes de serem apresentados aos utilizadores como informação relevante. Este capítulo encontra-se então dividido em duas secções, sendo que na primeira é abordada a obtenção de dados a partir das redes sociais e, na segunda, a obtenção de dados utilizando diversos sensores.

2.1 Redes Sociais

Recentemente, com o surgimento das redes sociais, quantidades significativas de informação começaram a ser partilhadas em tempo real entre vários utilizadores e comunidades. As redes sociais têm, cada vez mais, emergido como meios de propagação de notícias e informações. Ninguém pode negar a influência das redes sociais durante a Primavera Árabe⁶, onde estas funcionaram como alicerces para a organização e difusão de uma série de protestos antigovernamentais, manifestações e rebeliões armadas que se espalharam pelo mundo islâmico no início do século XXI.

Deste modo, as informações partilhadas pelos utilizadores têm vindo a tornar-se uma importante fonte de dados. Nesta subsecção, o foco será na obtenção de informações relacionadas com eventos que perturbem o

⁶ https://en.wikipedia.org/wiki/Arab_Spring

normal funcionamento do tráfego automóvel, utilizando para isso as redes sociais. Assim, as soluções que serão apresentadas nesta secção utilizam diferentes métodos e redes sociais distintas para a obtenção de informações relacionadas com o trânsito. Desta forma é possível informar os utilizadores acerca das melhores vias a seguir e também quais as que devem ser evitadas.

Kathri [9], Alomari et al. [10], Gu et al. [11], Ribeiro et al. [12] e Zhang et al. [13] propuseram sistemas de monitorização do tráfego automóvel extraíndo para isso diversas informações de trânsito. Para a implementação destes sistemas foi utilizada a rede social Twitter⁷ como principal fonte de dados, permitindo assim a deteção de acidentes e engarrafamentos a partir dos *tweets* dos utilizadores. Os sistemas desenvolvidos extraem diversos tipos de informação, utilizando para isso os próprios utilizadores como sensores, não existindo a necessidade da criação de uma infraestrutura dedicada para esse fim. Para isso, os *tweets* dos utilizadores são analisados e filtrados, focando-se em diversos parâmetros: o texto do *tweet*, a sua geolocalização (se disponível), a data, a hora e o *user_id*. Os dados são então extraídos utilizando diversos métodos, que serão apresentados nos 5 parágrafos abaixo.

A solução apresentada por Kathri [9] consiste na criação de *bounding boxes*, baseando-se em dois pares de coordenadas. O primeiro par corresponde à coordenada mais a sudoeste e o segundo par à coordenada mais a nordeste (Figura 5). Todos os *tweets* em que as coordenadas de geolocalização se encontrem dentro destas *bounding boxes* são armazenadas na base de dados. Posteriormente os *tweets* que possuam a palavra-chave *traffic* são filtrados e armazenados noutra base de dados. Por fim, são removidos todos os *tweets* que possuam a palavra-chave *traffic* mas que não se encontrem diretamente ligados com o tráfego automóvel (p. ex. *human trafficking*, *trafficker*, *air traffic*...) ou que tenham sido gerados por serviços profissionais de monitorização de trânsito.

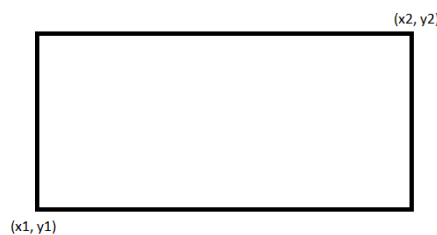


Figura 5: Coordenadas necessárias para a criação de uma *bounding box* [9].

⁷ <https://twitter.com/>

Alomari et al. [10] também apresentaram uma solução baseada na geolocalização dos *tweets* para obter eventos relacionados com o tráfego automóvel, mas ao invés de utilizar *bounding boxes* basearam-se em todos os *tweets* que se encontrassem dentro de um determinado raio. A ideia é delimitar uma distância de tolerância em torno de um determinado ponto para, posteriormente, validar se as coordenadas do *tweet* se encontram dentro do raio delimitado. De modo a filtrar os dados recolhidos por esta plataforma foi criado um dicionário contendo uma lista de palavras-chave (p. ex. *accident, fire, roadworks...*), todas elas relacionadas com eventos que pudessem condicionar o tráfego automóvel. Neste dicionário foram também inseridas palavras-chave relacionadas com as condições do tempo (p. ex. *rain, storm...*), com eventos desportivos e com eventos religiosos. O sistema foi desenvolvido utilizando o *software* SAP HANA⁸, que é uma plataforma que permite recolher, armazenar e processar grandes volumes de dados em tempo real, possuindo a opção de executar a recolha de dados de forma automática a partir de REST APIs.

O método utilizado por Gu et al. [11] consiste também na filtragem e análise dos *tweets* que possuam determinadas palavras-chave (p. ex. *traffic, accident, road, avenue, car, driver, congestion...*) baseando-se para isso num dicionário de palavras-chave. Para tornar este dicionário o mais completo possível foi utilizada uma estratégia chamada de *Adaptive Data Acquisition* que permite, em cada iteração, adicionar novas palavras ao dicionário tendo como base as palavras-chave iniciais. Inicialmente este dicionário possuía apenas 50 palavras-chave, sendo que após várias iterações este dicionário aumentou para um total de 131 palavras-chave. De mencionar que ao utilizar esta estratégia também são adicionadas palavras-chave ao dicionário que não são relacionadas com o tráfego automóvel, o que gerou cerca de 383 falso-positivos. Outras fontes importantes de informação utilizadas foram as contas do Twitter de várias entidades diretamente relacionadas com o trânsito, tais como agências do Departamento de Transportes e canais de notícias, que permitiram contrastar com os dados obtidos a partir dos *tweets* dos utilizadores.

Na solução apresentada por Ribeiro et al. [12] foram filtrados e analisados todos os *tweets* que possuíssem o nome de alguma via, sido também tidas em conta todas as abreviações (p. ex. Fleming Av.). Este método consiste no pressuposto de que a grande maioria dos *tweets* em

⁸ <https://www.sap.com/products/technology-platforms/database-management-system.html>

que alguma via é mencionada são, de alguma forma, relacionados com algum evento no trânsito, quer sejam engarrafamentos ou acidentes. No fundo este sistema segue a ideia de que os utilizadores tendem a partilhar os eventos que ocorram ao seu redor com outras pessoas, assim como os eventos que estejam diretamente relacionados com o tráfego automóvel. Foram também recolhidos *tweets* de entidades cujo objetivo é fornecer informações acerca das condições do trânsito, sendo que estas incluíam perfis que partilhavam informações de trânsito de forma voluntária, estações de rádio, empresas de táxi e também a conta oficial do Departamento de Trânsito da cidade. Por fim, os dados obtidos a partir destas contas eram contrastadas com os dados obtidos através das recolhas iniciais, que eram mais genéricas e cujo principal objetivo não era fornecer este tipo de informações.

Zhang et al. [13] apresenta uma solução para detetar eventos importantes que aconteçam e que possam afetar o tráfego automóvel, tendo em conta a quantidade de *tweets* partilhados contendo uma palavra-chave (nome do evento) num determinado local. Quanto maior o número de *tweets* que possuam essa palavra-chave, maior é a probabilidade de um evento se encontrar a acontecer nesse local. A diferença entre esta solução e as anteriores é que nesta solução a preocupação não se encontra em obter *tweets* com determinadas palavras-chave, mas sim saber a quantidade de vezes que, num determinado local, o nome de um evento apareceu nos *tweets* dos utilizadores. Ao comparar estas concentrações com os padrões de tráfego em torno dos *tweets* foi determinado que em 77,4% das vezes estas levaram ao aumento do tráfego local. Alguns dos eventos detetados pela solução apresentada foram: a abertura de cerimónias, a morte de celebridades, festivais e conferências internacionais. Eventos estes que se tornavam problemáticos devido à aglomeração de pessoas num determinado local e também no aumento do tráfego automóvel.

Por outro lado, Chatterjee et al. [14] e Akilesh et al. [15] desenvolveram interfaces *online* que permitem a recolha de dados em tempo real sobre o tráfego das cidades indianas a partir de páginas da polícia de trânsito no Facebook (Figura 6), utilizando para isso a API do Facebook. Estas permitem extrair diferentes atributos de forma dinâmica, dando assim a possibilidade de os autores explorarem o estado e a qualidade do tráfego em tempo real. Assim, foi possível filtrar os *posts* dos utilizadores, de forma a apenas recolher os dados relacionados com eventos que pudessem influenciar o trânsito. De forma semelhante, a API

acima mencionada será posteriormente utilizada para a plataforma que foi desenvolvida, de modo a ser possível a obtenção de informações acerca do trânsito a partir dos *posts* dos utilizadores.

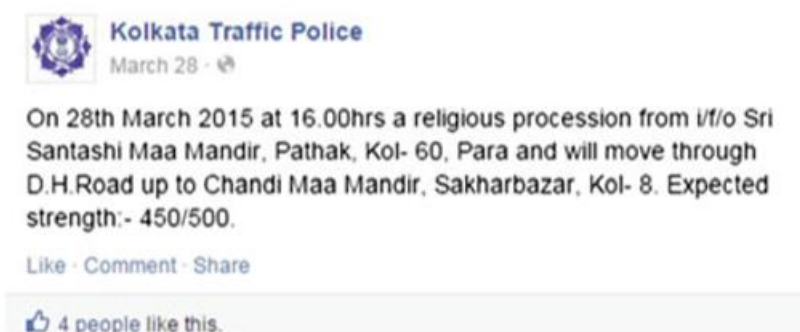


Figura 6: Exemplo de um post na página da polícia de trânsito de Calcutá indicando que ocorrerá uma procissão religiosa que irá obrigar ao encerramento de algumas ruas [14].

Outros estudos basearam-se na rede social Weibo⁹ (equivalente ao Twitter na China) para a obtenção de dados relacionados com o tráfego. Na solução apresentada por Cui et al. [16] foram desenvolvidos dois métodos para a obtenção dos dados, sendo que no primeiro existe uma filtragem das mensagens partilhadas pelos utilizadores enquanto que no segundo é realizada uma pesquisa por determinadas *hashtags* relacionadas com eventos que influenciem o tráfego automóvel. Já Pan et al. [17] focaram-se na obtenção de dados acerca de acidentes, protestos, eventos desportivos, celebrações e desastres, baseando-se na deteção de mudanças significativas nas rotas dos utilizadores. Para isso, quando é detetado um comportamento anómalo nas rotas normalmente utilizadas pelos utilizadores, é procurado um contexto semântico para dar sentido a essa mudança, ou seja, aproveitam-se das redes sociais para procurar encontrar descrições para essas anomalias. Nesse caso, existe um correlacionamento entre as interrupções nos padrões de tráfego com a sua semântica, ou seja, os eventos que obrigam os utilizadores a procurarem novas rotas.

2.2 Sensores

A utilização de sensores para obter informações de tráfego em tempo real tem-se disseminado cada vez mais, seja pelo baixo custo de aquisição destes equipamentos, seja pela facilidade em obter grandes volumes de

⁹ <https://www.weibo.com/overseas>

dados, ou até mesmo pelo seu tamanho. Entretanto, tem sido um desafio determinar onde estes equipamentos devam ser localizados para maximizar os benefícios da sua utilização, e também a forma como deverão ser tratados estes volumes de dados de modo a fornecer um tipo de informação qualitativa, ao invés de quantitativa.

Singh et al. [18], Allouch et al. [19] e Bhoraskar et al. [20] apresentam diferentes soluções para detetar as condições da superfície das estradas. Singh et al. [18] utilizam uma técnica de *crowdsourcing* e de utilização dos sensores presentes nos dispositivos móveis para detetar as condições da superfície das estradas, como por exemplo, buracos e lombas, com base na análise das leituras do acelerómetro. Foi desenvolvida também uma aplicação Android de modo a poder interagir com os sensores dos dispositivos móveis, coletando deste modo os dados do acelerómetro e do GPS. Estes *smartphones* foram colocados dentro de um veículo em posições diferentes (painel frontal ou banco de trás), e o condutor ao passar por um buraco ou lomba, faria com que o acelerómetro detetasse essas variações na superfície da estrada.

O RoadSense [19], baseado na aplicação The Pothole Patrol [21], é uma plataforma de monitorização das condições das vias, baseando-se para isso nos sensores presentes nos *smartphones* (acelerómetro, GPS e magnetómetro). Esta plataforma, tal como a aplicação Wolverine [20], permitem também detetar o tráfego automóvel identificando os eventos relacionados com a travagem do automóvel, que frequentemente indicam que o trânsito se encontra congestionado. A plataforma aplica também técnicas de *machine learning* para classificar os dados de modo a se poder adaptar a diferentes fatores, como a natureza da estrada e o tipo de veículo.

Mazhar et al. [22] apresenta uma solução para determinar as condições das estradas no Paquistão. Para isso foi desenvolvida uma aplicação em Android que deteta buracos e lombas de limitação de velocidade que existam na via. As informações são enviadas para um servidor, e posteriormente utilizadas pelas autoridades municipais para que estas possam reparar os buracos nas estradas. A aplicação consegue detetar com sucesso cerca de 90% dos buracos existentes nas vias e também cerca de 95% das lombas limitadoras de velocidade. A aplicação tem dois modos de obtenção destes dados, um deles sendo através de sensores (nomeadamente o GPS e três tipos de acelerómetro) que permitem a obtenção dos dados brutos que posteriormente são tratados de forma a que estes possam ser apresentados ao utilizador, e também

através de informações fornecidas pelos próprios utilizadores, que através de uma funcionalidade presente no mapa da aplicação podem adicionar a localização de buracos na estrada e enviar estes dados para todos os outros utilizadores da aplicação. Neste último caso a aplicação guarda as coordenadas GPS e o *timestamp* de modo a que posteriormente a localização deste obstáculo na via possa ser apresentada na aplicação.

Similarmente à plataforma desenvolvida, esta aplicação também utiliza os utilizadores como uma mais-valia para a obtenção de informações relacionadas com as condições das vias e também das condições de trânsito. Para além disso utiliza também os sensores de modo a automatizar de algum modo a obtenção destas mesmas informações. De salientar que se torna sempre necessária a filtragem destes dados em bruto, de modo a que estes possam ser fornecidos aos utilizadores como informações relevantes.

2.3 Conclusões

Devido ao enorme volume de dados gerado diariamente nas redes sociais não é, portanto, incomum encontrar diversos estudos que apontem as redes sociais como possíveis fontes de dados para soluções que tentem combater os cada vez mais preocupantes congestionamentos de trânsito. Assim, o Twitter, o Facebook, o Weibo e outras redes sociais podem agir como novas plataformas para a obtenção de dados, onde os próprios utilizadores agem como sensores, compartilhando assim a informação que eles possuem. Esta nova abordagem baseada nas redes sociais para a monitorização do congestionamento do tráfego automóvel permite que, para além dos condutores, também os pedestres, ciclistas e passageiros possam ser utilizados como sensores de onde podemos obter informações valiosas [23].

Outra plataforma que utiliza diversas fontes para poder informar os utilizadores acerca de estradas fechadas ou condicionadas é o *website* do Serviço Regional de Proteção Civil da Madeira¹³. Para além do *website* estes também possuem uma aplicação denominada ProCiv Madeira¹⁴ que apresenta basicamente os mesmos dados. O procedimento utilizado pelo Serviço Regional de Proteção Civil da Madeira para a obtenção destes dados baseia-se na pesquisa de editais que são publicados em formato

¹³ <https://www.procivmadeira.pt/pt/>

¹⁴ <https://play.google.com/store/apps/details?id=com.dobsware.procivmadeira>

PDF pelos diversos municípios da RAM. A informação publicada nestas duas plataformas centraliza de algum modo toda a informação municipal relacionada com interrupções do tráfego automóvel.

A plataforma a ser apresentada foi desenvolvida a partir de fontes de dados não tradicionais, utilizando para isso os dados do *website* do Serviço Regional de Proteção Civil da Madeira e também os dados provenientes do grupo Ocorrências Madeira¹⁶ existente na rede social Facebook. Para atingir este objetivo, primeiramente foram utilizados dados que possuíssem informações de geolocalização, de modo a que pudessem ser representados no mapa da aplicação Android.

Em relação à utilização de sensores para detetar as condições da superfície das estradas e tendo em conta os estudos apresentados anteriormente, a plataforma apresentada utiliza os dados obtidos a partir de *posts* no Facebook acerca das condições da via e outras anomalias graves na superfície das estradas. Tal como nos estudos apresentados anteriormente, muitos dos utilizadores compartilham informações valiosas relacionadas com o tráfego automóvel, o que permite com que a plataforma possa identificar onde ocorreram acidentes ou congestionamentos de trânsito. Por outro lado, obtém também informações sobre quedas de rochas e deslizamentos de terra, problemas que se estendem a toda a ilha, principalmente no Inverno. A plataforma desenvolvida não utiliza nenhum método para detetar as condições da superfície das estradas, mas é algo que futuramente poderá ser integrado.

¹⁶ <https://www.facebook.com/groups/ocorrencias.madeira/>

3. Arquitetura de Software

A arquitetura da plataforma desenvolvida baseia-se numa arquitetura de três camadas (*3-tier*), que são frequentemente utilizadas em arquiteturas cliente-servidor. Esta arquitetura permite diferenciar em 3 camadas os 3 ambientes de desenvolvimento mais comuns, nomeadamente a interface do utilizador, a lógica funcional e as bases de dados ou quaisquer outros armazenamentos de dados.

As vantagens de uma arquitetura de três camadas (*3-tier*), baseiam-se principalmente na separação de responsabilidades, em que será possível a obtenção de uma certa independência entre cada uma das implementações. Esta arquitetura é também, devido à separação das diferentes camadas, de fácil modificação. Como cada camada é independente das outras, é possível a atualização ou alteração de uma das camadas sem afetar a plataforma como um todo. Por exemplo, seria possível a implementação de diferentes bases de dados, não ficando assim bloqueados numa tecnologia específica. Sendo assim, esta arquitetura permite também uma maior gestão do código, uma vez que a alteração do código de uma das camadas não influencia diretamente nenhuma das outras duas camadas.

Outros benefícios em utilizar uma arquitetura *3-tier* são: a velocidade de desenvolvimento, a escalabilidade, o desempenho e a disponibilidade. A modularização das diferentes camadas de uma plataforma oferece a capacidade de desenvolver e aprimorar os diferentes componentes de uma forma mais rápida e “limpa”, uma vez que é aplicada a separação de responsabilidades, o que também ajudará futuramente caso tenhamos de modificar algo na plataforma. Neste caso a atualização de uma camada específica, terá um impacto mínimo nas outras camadas. Portanto, esta modularização irá também ajudar a recuperar de um desastre inesperado, permitindo aos desenvolvedores se concentrar apenas no módulo que tenha apresentado erros.

A escalabilidade é outra grande vantagem desta arquitetura, pois ao separar os diferentes componentes da plataforma, permite realizar o redimensionamento de cada um destes de forma independente. Isto também permite mudar o tipo de tecnologias utilizadas, caso seja decidido modificar o tipo de base de dados, o tipo de servidor ou até mesmo o tipo de plataforma móvel a ser utilizada. Deste modo será possível, por exemplo, aumentar o espaço de armazenamento da base de dados sem mudar obrigatoriamente as características do servidor.

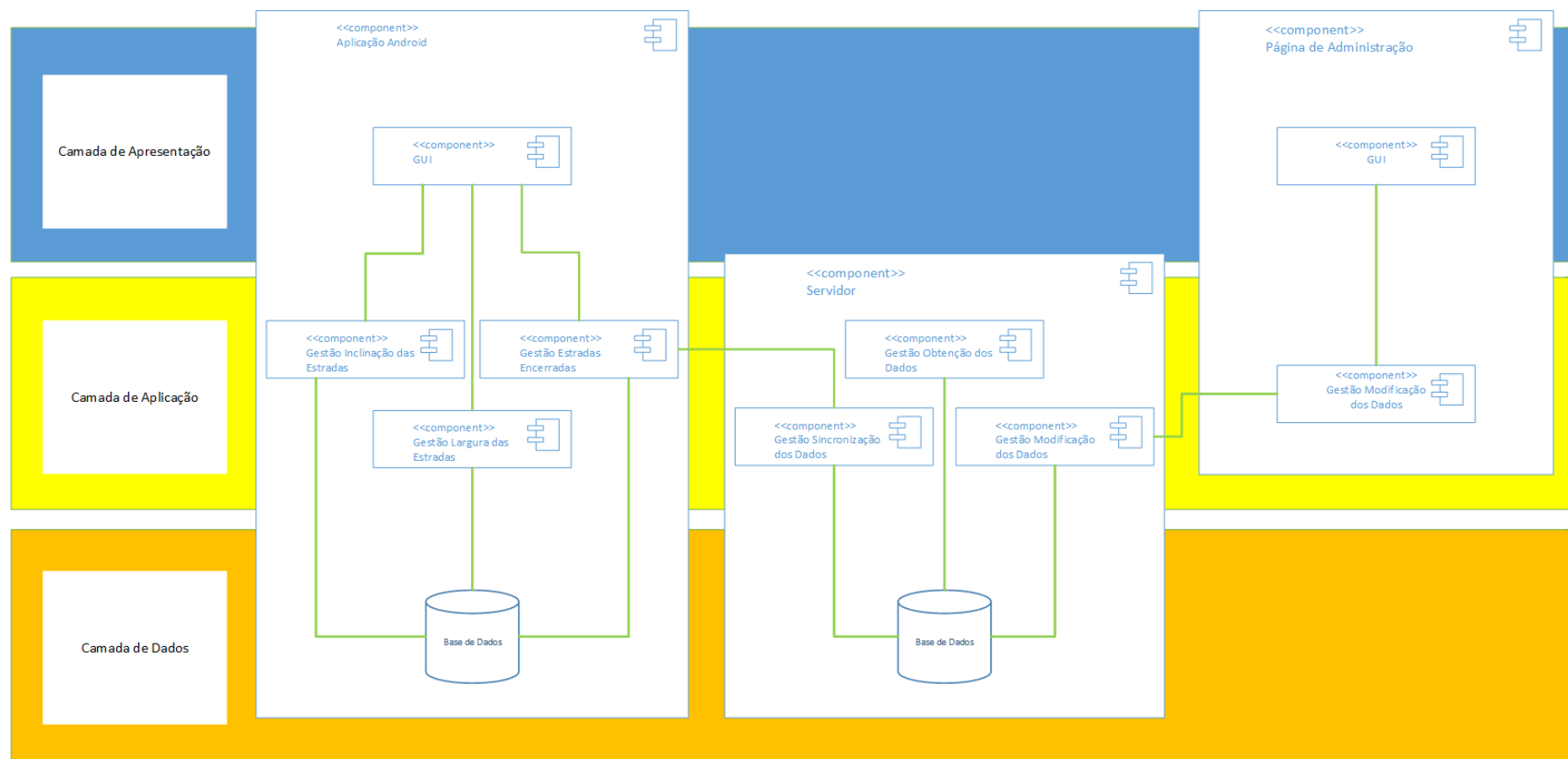


Figura 7: Arquitetura da Plataforma.

Este tipo de modificabilidade, não do código, ou dos módulos, mas sim das tecnologias, foi tida em conta utilizando como servidor o Node.js, que é conhecido por se integrar bem com qualquer DBMS e também o MySQL, reconhecida por se integrar bem com qualquer tipo de servidor. No caso do Node.js este suporta a maioria dos tipos de bases de dados¹⁷, quer estas sejam bases de dados relacionais (MySQL, MS SQL Server, Oracle, PostgreSQL, SQLite...) ou NoSQL (MongoDB, Cassandra, LevelDB, RavenDB, Neo4j, Redis, CouchDB...). Já no caso de ter sido optado pela base de dados relacional MySQL, esta primeiramente é suportada pela maioria dos sistemas operativos¹⁸, permitindo assim que seja possível mudar de plataforma de maneira mais fácil e segura.

Ao ser utilizada uma arquitetura modular, que permite ter componentes facilmente escaláveis, será deste modo mais fácil conseguir com que todos os serviços estejam disponíveis a maior parte do tempo. Uma prova disto é que caso algum dos componentes não esteja a funcionar, todos os restantes se mantêm operacionais. Um bom exemplo disto seria caso o servidor fosse abaixo, a aplicação Android continuaria a funcionar normalmente, sem interrupções e vice-versa.

A arquitetura *3-tier* (Figura 7) divide-se nestas 3 camadas:

- Camada de Apresentação – Corresponde à interface do utilizador (GUI). Nesta camada podemos encontrar as duas interfaces de utilizador a serem implementadas, nomeadamente a página de administração, que será construída através de uma interface *web* e a própria GUI da aplicação Android.
- Camada de Aplicação – Corresponde aos componentes onde será implementada a lógica funcional, ou seja, todas as ações responsáveis para que o sistema possa funcionar como pretendido.
- Camada de Dados – Corresponde aos componentes onde serão armazenados os dados. Neste caso existirá uma base de dados no servidor e também na própria aplicação Android, que permitirá o armazenamento dos dados diretamente nos dispositivos, e que irá replicar as informações existentes na base de dados do servidor.

Tendo em conta a arquitetura apresentada, a plataforma irá dividir-se então em 3 grandes componentes todos eles pertencentes às camadas acima mencionadas. Uma vez que tanto o servidor como a aplicação

¹⁷ <https://www.tutorialsteacher.com/nodejs/data-access-in-nodejs>

¹⁸ <https://www.mysql.com/support/supportedplatforms/database.html>

Android possuem bases de dados próprias, a Camada de Dados estará encapsulada dentro de cada um dos componentes:

- Servidor – Onde será feita toda a gestão relacionada com a obtenção, filtragem, armazenamento e envio dos dados para os dispositivos móveis. Esta é a plataforma central da aplicação conjugando o servidor e a base de dados.
- Aplicação Android – Onde serão mostrados os dados aos utilizadores, obtendo-os primeiramente a partir de diversas fontes (servidor e fontes externas) e armazenando-os diretamente em bases de dados locais.
- Página de Administração – Onde o administrador de sistema poderá modificar ou eliminar alguns dos dados obtidos a partir do Serviço Regional de Proteção Civil da Madeira, de forma manual.

Abaixo serão apresentados alguns diagramas de sequência que exemplificam as interações entre estes três componentes presentes na plataforma.

Posteriormente serão também apresentados os diferentes atributos de qualidade ordenados por relevância com os respetivos cenários de qualidade.

3.1 Interação entre os Componentes

Para representar a interação entre os diferentes componentes foram utilizados diversos diagramas de sequências. Uma das razões que levaram à utilização deste tipo de diagramas foi a facilidade de demonstrar a lógica de interação entre os componentes bem como a sequência temporal em que essas interações ocorrem. Outra das razões para a utilização de diagramas de sequências é a importância inerente à obtenção de dados externos à plataforma (uma vez que a plataforma precisa de consumir dados e serviços para que consiga atingir os seus objetivos), permitindo assim representar de uma forma simples e lógica a obtenção desses mesmos dados.

Os diagramas de sequências são diagramas UML que ilustram a sequência de mensagens transmitidas entre diversos objetos numa interação, dando ênfase à sequência temporal em que essas mensagens são trocadas entre os objetos de um sistema. Um diagrama de sequências é no fundo estruturado de forma a representar uma linha do tempo, em que as primeiras interações se encontram no topo, descendo gradualmente

à medida que novas interações são realizadas entre os diversos componentes.

3.1.1 Tratamento dos Dados obtidos a partir do Serviço Regional de Proteção Civil da Madeira

O Anexo A é composto pelo diagrama de sequências onde é representado o tratamento dos dados obtidos a partir do Serviço Regional de Proteção Civil da Madeira, focando na obtenção dos dados obtidos a partir desta mesma fonte.

Os dados são obtidos a partir do *website* da referida entidade, que centraliza a informação presente em editais que são publicados em formato PDF pelos diversos municípios da RAM. A informação publicada nesta plataforma centraliza de algum modo toda a informação municipal relacionada com interrupções do tráfego automóvel, mas também outras informações que sejam relevantes para a população. Uma vez que a informação relevante já se encontra de algum modo normalizada, ou seja, tratada, torna-se assim mais fácil a obtenção deste tipo de informações, evitando que fosse necessária a obtenção destes dados a partir de cada *website* camarário. Outra vantagem em utilizar este *website* como fonte, é o tratamento prévio de todas as informações presentes nos editais e a utilização de um mapa do Google Maps para a apresentação destas informações, e onde é possível obter as coordenadas aproximadas para o encerramento ou condicionamento das vias em questão.

Abaixo serão explicados os diversos passos para a obtenção e tratamento destes dados, bem como as diferentes interações entre os diversos componentes da plataforma. De salientar que o acesso e obtenção destes dados é inicialmente realizado a partir do servidor Node.js, em que é feito um pedido para o *website* do Serviço Regional de Proteção Civil da Madeira.

De início são obtidos os dados já existentes no *website*, mas que se encontram em KML, que inicialmente era um formato de ficheiros utilizado para mostrar dados geográficos no Google Earth, mas que posteriormente foi também adaptado para utilizar no Google Maps. Este formato baseia-se no formato XML e permite o seu *parsing* utilizando qualquer *parser* XML. De modo a uniformizar os *parsings* e tendo em conta que os dados devolvidos por todos os outros serviços se encontram

todos em JSON, foi decidido passar o KML para objetos JSON e assim realizar o *parsing* neste formato.

Através deste ficheiro são obtidos diversos dados acerca das vias encerradas, sendo que o próximo passo consiste em verificar se a informação acerca de cada via já se encontra na base de dados ou não. Caso já exista, a informação referente a essa via é descartada, caso ainda não exista, é obtido o nome da via encerrada a partir das coordenadas obtidas. Para isso foi utilizado um dos serviços fornecidos pelo Overpass API¹⁹ (API otimizada para a obtenção em grande escala de dados provenientes do OSM, existindo a possibilidade de utilizar critérios de pesquisa, através do Overpass QL, tais como a localização, o nome e tipo de vias, os tipos de serviços existentes nas proximidades...) que permite obter o nome de estradas, tendo como parâmetros de pesquisa um par de coordenadas e o raio (parâmetro *around*). O raio passado como parâmetro foi cuidadosamente testado, pois ao fornecermos um raio muito grande, acabamos por obter o nome das estradas circundantes, e se fornecermos um raio muito pequeno podemos não obter o nome de nenhuma via. Após diversos testes foi decidido que o raio a ser passado seria de 20 metros.

Posteriormente, são obtidos todos os nós (coordenadas) que compõem uma via, passando para isso, como parâmetro, o nome da via obtido anteriormente (utilizando também um dos serviços do Overpass API).

Utilizando o serviço Directions API da MapQuest (Find Link Id – GET²⁰), são obtidos os Link Ids das coordenadas existentes para cada via, que serão posteriormente utilizados como valores do parâmetro `mustAvoidLinkIds` nas rotas traçadas na aplicação. Estes valores permitirão posteriormente traçar rotas evitando assim estes troços de estrada que se encontram encerrados. A MapQuest disponibiliza assim uma forma de podermos traçar estas rotas evitando determinados troços de estrada, sendo que, no entanto, existe a necessidade da obtenção destes Link Ids, uma vez que não existe a possibilidade de passar os valores das coordenadas nos serviços de navegação.

Por fim estes dados são inseridos na tabela `temp_civil_protection` com a *flag* editado a *true* e posteriormente na tabela `civil_protection` caso os dados se encontrem normalizados, ou seja, com todos os dados corretos (principalmente a data e hora de encerramento e a data e hora de

¹⁹ <http://www.overpass-api.de/>

²⁰ <https://developer.mapquest.com/documentation/directions-api/find-link-id/get/>

reabertura). Caso não se encontrem normalizados, a *flag* editado é colocada a *false* indicando que necessitará de intervenção humana para a sua normalização. Neste caso é enviada uma notificação para o *email* do administrador de sistema informando que existe a necessidade da sua intervenção.

3.1.2 Normalização dos Dados na Página de Administração

No Anexo B é representada a normalização dos dados obtidos a partir do Serviço Regional de Proteção Civil da Madeira, normalização que é feita na página de administração. Esta página foi criada principalmente devido à necessidade de normalização dos dados, para que estes possam ser utilizados na plataforma. As iterações do administrador de sistema nesta página e posterior atualização dos dados na base de dados são representados utilizando para isso um diagrama de sequências.

Esta plataforma apenas permite o seu acesso através de uma página de *login* onde o administrador de sistema deverá introduzir as suas credenciais. Após o administrador de sistema introduzir corretamente os dados de *login* este será redirecionado para a página onde será possível editar ou eliminar os dados das vias encerradas, informação obtida a partir do Serviço Regional de Proteção Civil da Madeira.

No caso de o administrador editar os dados, existem validações para que os dados sejam apenas guardados se este introduzir corretamente todos os dados pretendidos. Neste caso, após a correta validação dos dados estes são atualizados na tabela *temp_civil_protection* colocando a *flag* editado a *true* e posteriormente o registo é também guardado na tabela *civil_protection*.

No caso de o administrador eliminar os dados, estes não são realmente excluídos, sendo apenas colocada a *flag* editado a *true* na tabela *temp_civil_protection*. Neste caso, este registo não é mais mostrado na página de administração, pois apenas são apresentados nesta página os registos em que *flag* editado se encontre a *false*. Esta estratégia foi adotada tendo em conta que após a eliminação dos dados se torna difícil a sua recuperação. Poderíamos também realizar a recuperação dos dados através dos ficheiros de *binlog*, mas de forma a não correr riscos e devido a más experiências noutras plataformas foi decidida a implementação desta estratégia.

3.1.3 Envio dos Dados para a Aplicação

No Anexo C é representado o diagrama de sequências dos eventos que irão desencadear o envio dos dados provenientes do Serviço Regional de Proteção Civil da Madeira para a aplicação móvel. Os eventos de inserção, atualização e eliminação de registos na tabela *civil_protection* são comandos DML que irão desencadear a sua replicação num ficheiro *binlog*, que no fundo nada mais é que um *log* de todos os comandos executados na base de dados.

Cada vez que é executado um comando DML, o ficheiro *binlog* é atualizado de forma a conter uma replicação exata dos dados existentes na base de dados. No fundo funciona como uma espécie de *journaling*, presente em outras DBMS, permitindo o registo simultâneo (em tempo real) de todas as alterações numa tabela ou até mesmo em toda a base de dados. Este *log* é utilizado normalmente para realizar a recuperação de uma tabela ou até mesmo de toda a base de dados, caso os dados tenham sido danificados ou destruídos.

Cada uma das mudanças nos ficheiros de *log* despoleta um *trigger* no servidor, que então detetará que tipo de mudança foi executada nos dados e caso necessário executa uma série de comandos de forma a atualizar os dados na aplicação móvel.

Caso o comando executado na tabela *civil_protection* tenha sido um INSERT, um UPDATE ou um DELETE, é realizado o *broadcast* dessa mudança para os dispositivos móveis, enviando na mensagem um objeto JSON com os novos dados no caso do INSERT, com os dados a serem modificados no caso do UPDATE e com o *id* do registo no caso do DELETE. Para o envio destas mensagens foi utilizado o Socket.io²¹, que nada mais é do que uma implementação do protocolo de transporte em tempo real, permitindo a comunicação bidirecional entre cliente e servidor. Nesta implementação este foi utilizado de modo a enviar *push notifications* para todos os clientes, sempre que exista uma alteração relativamente aos dados presentes na tabela *civil_protection*. Para isso foi utilizada umas das suas funcionalidades, a possibilidade de fazer *broadcasting*, bem como a de poder armazenar os dados associados a cada cliente. Esta última foi utilizada principalmente para saber o *timestamp* do último acesso do cliente ao servidor, permitindo assim enviar todos os

²¹ <https://socket.io/>

dados que este não tenha recebido devido a não se encontrar conectado à *Internet* no momento de algum *broadcast*.

Por fim, na aplicação móvel é realizado um *parsing* destes objetos, executando posteriormente o comando DML que é passado na mensagem. Estes dados são guardados nas bases dados locais de cada aplicação móvel e mostrados posteriormente no mapa da aplicação.

3.2 Atributos e Cenários de Qualidade

Entre as características necessárias para uma boa arquitetura de *software* estão a separação de responsabilidades, já referida anteriormente em relação aos diferentes componentes da arquitetura, e esta ser orientada por determinados atributos de qualidade. Não são apenas estas características que são importantes para uma boa arquitetura de *software*, mas estas são as que facilmente podem ser identificadas ao ser utilizada uma arquitetura *3-tier*. É sempre importante definir os atributos de qualidade mais importantes para o sistema a ser construído, tendo em vista não só a funcionalidade do sistema em si, mas também a sua tolerância a falhas, a compatibilidade com outros sistemas dos quais dependamos, a confiabilidade, a capacidade e facilidade de manter a plataforma correndo sem erros, a disponibilidade, a segurança... Neste sistema foram definidos como atributos de qualidade de maior importância: o desempenho, a modificabilidade, a disponibilidade, a interoperabilidade e a segurança. Isto não invalida que a implementação de todos os outros atributos de qualidade não tenha sido feita, mas sim que houve um maior foco nestes cinco. A Tabela 4 apresenta os atributos de qualidade ordenados por relevância, incluindo também os respetivos cenários de qualidade.

Nível de Prioridade	Atributo de Qualidade	Cenário de Qualidade
1	Disponibilidade	A aplicação deverá conseguir obter a informação disponível no servidor logo que esta tenha acesso à <i>Internet</i> . O servidor deverá conseguir obter os dados provenientes das fontes citadas posteriormente e também estará acessível a todos os dispositivos móveis que possuam a aplicação instalada.
2	Segurança	A aplicação deverá garantir a confidencialidade e a segurança na transmissão dos dados. O servidor e a base de dados apenas deverão ser acedidos pelo administrador de sistema através

		da utilização de credenciais, e os dados deverão ser alterados com as devidas permissões.
3	Desempenho	A aplicação deverá conseguir obter os dados a partir do servidor no mínimo tempo possível.
4	Modificabilidade	Todos os componentes da plataforma devem ser facilmente modificáveis. A plataforma deverá permitir a modificação da base de dados sem necessitar de modificar a lógica de toda a plataforma. A plataforma deverá ser de fácil manutenção tanto na questão da atualização das diversas tecnologias utilizadas, como ao <i>bug fixing</i> de erros que possivelmente poderão aparecer.
5	Interoperabilidade	A aplicação deverá poder trocar mensagens com os serviços externos (servidor, serviços da Google, MapQuest...) com uma taxa de sucesso superior a 80%.

Tabela 4: Cenários de qualidade ordenados por prioridade.

Os atributos de qualidade com maior nível de prioridade serão aqueles em que irá existir uma maior atenção, de modo a que a aplicação possa estar disponível numa grande percentagem do tempo, mas que também possa ao mesmo tempo fornecer ao utilizador final a privacidade e a segurança dos seus dados.

O atributo disponibilidade, tem em conta não só a disponibilidade dos serviços fornecidos pela aplicação, mas também a possibilidade de o servidor obter através de outros serviços e fontes de dados a informação a guardar e a enviar para os utilizadores. Será tida também em conta a redundância dos dados, procurando de algum modo manter um repositório local dos dados na aplicação. Este repositório deverá fornecer os dados à aplicação caso não exista ligação à Internet ou caso não exista ligação com o servidor. Quando exista ligação à Internet estes dados devem ser atualizados de imediato, conectando-se deste modo com o servidor ou com os serviços externos que serão posteriormente mencionados.

O atributo segurança, devido ao Regulamento Geral de Proteção de Dados implementado pela União Europeia, irá ser utilizado nomeadamente na implementação da encriptação de dados sensíveis na base de dados, mas também pela utilização do protocolo HTTPS para o envio dos dados do servidor para a aplicação. Neste caso, os dados dos pedidos ao servidor devem ser encriptados, se possível, utilizando SSL

através do protocolo HTTPS, protegendo assim de ataques do tipo *man-in-the-middle*.

Para o atributo desempenho, serão analisados muitos dos *requests* realizados aos vários serviços utilizados pela plataforma, bem como os acessos a serviços externos de onde são obtidos os dados que serão apresentados na aplicação.

O atributo modificabilidade, será também um dos aspetos fundamentais desta plataforma. Tanto no servidor, na base de dados ou na aplicação o código deverá ser implementado de modo a ser facilmente modificável, utilizando para isso diversos módulos, separando assim as responsabilidades de cada módulo. Outro dos aspetos a ter em conta, como mencionado anteriormente, será a fácil mudança de tecnologias entre os diversos componentes da arquitetura de sistema, de forma a que exista a possibilidade de a plataforma poder crescer e melhor se adaptar às tecnologias mais recentes no mercado, evitando assim a utilização de tecnologias que se possam tornar obsoletas. Para além disso, por ser uma arquitetura *3-tier*, esta deverá também, inerentemente à utilização de três camadas onde se encontram os componentes, de ser de fácil modificabilidade e de ser de fácil manutenção tanto à atualização dos seus elementos como ao *bug fixing* de erros.

Por fim, o atributo interoperabilidade é importante devido, a como foi acima descrito, à vasta utilização de serviços e à obtenção de dados externos. Deste modo é procurado que as comunicações entre os diferentes componentes da plataforma sejam feitas da forma mais transparente possível, e que estas comunicações sejam feitas sem falhas, evitando assim problemas que possam afetar severamente a plataforma.

4. Desenvolvimento da Aplicação

Neste capítulo será descrito o processo de desenvolvimento do sistema com os seus diversos componentes e também com as diversas funcionalidades que foram adicionadas à solução apresentada.

Esta secção encontra-se então dividida em 6 subsecções sendo elas: a plataforma de administração onde é apresentada a página *web* onde o administrador de sistema poderá gerir os dados provenientes do Serviço Regional de Proteção Civil da Madeira, a implementação do DNS no servidor, a implementação do protocolo HTTPS no servidor, o servidor Node.js e todos os módulos que o compõem, a base de dados existente no servidor e por fim a plataforma Android.

4.1 Plataforma de Administração

A plataforma de administração foi construída de modo a possibilitar que os administradores de sistema pudessem normalizar os dados, sem terem de aceder diretamente ao SQL. Deste modo garantimos a disponibilidade, dentro do possível de toda a informação proveniente do Serviço Regional de Proteção Civil da Madeira. A informação é normalizada uma vez que, muitas das vezes os dados obtidos não vêm com a data introduzida num formato *date*, e outras vezes a informação simplesmente não é colocada na plataforma, mesmo quando esta existe nos editais emitidos pelas entidades municipais.

Estas páginas foram desenvolvidas utilizando o *package* XAMPP²². Este *package* consiste essencialmente na base de dados MySQL, no servidor HTTP Apache e nos interpretadores para linguagens de *script*: PHP e Perl. Para além de ser *cross-platform* este *package* é também *open source*, sendo estas duas das principais vantagens que fizeram com que a implementação fosse feita utilizando este conjunto de plataformas.

Para a implementação das páginas em si foi utilizada a *framework* Bootstrap²³, que também é *open-source*, permitindo o rápido desenvolvimento do *front-end* de *websites* e aplicações, para isso tirando partido de diferentes linguagens, tais como: HTML, CSS e JavaScript. A escolha desta *framework* deveu-se a esta possuir um *design* que permitia uma fácil e rápida implementação da plataforma de administração, para além de

²² <https://www.apachefriends.org/index.html>

²³ <https://getbootstrap.com/>

que possibilitava a criação de *responsive websites* permitindo assim que o administrador de sistema pudesse aceder à plataforma através de qualquer dispositivo móvel.

A plataforma de administração divide-se em duas páginas: a página de *login* (index.php) e a página principal (tables.php). Na página de *login* (Figura 8) o administrador de sistema realiza o *login* utilizando para isso as suas credenciais (*email* e *password*). A verificação das credenciais é efetuada acedendo à tabela *users_login* da base de dados *users*. Para gerar e verificar a *password* é utilizado o algoritmo de *hashing* *bcrypt*²⁴. Este por sua vez utiliza o algoritmo Eksblowfish para realizar o *hash* das *passwords*. Este algoritmo garante que após o *hash* da *password*, não seja possível a sua recuperação sem o conhecimento do *salt*, do *cost*, e da chave (*password*). Deste modo é implementado o requisito de segurança nos acessos dos administradores de sistema à plataforma, neste caso à página de administração.

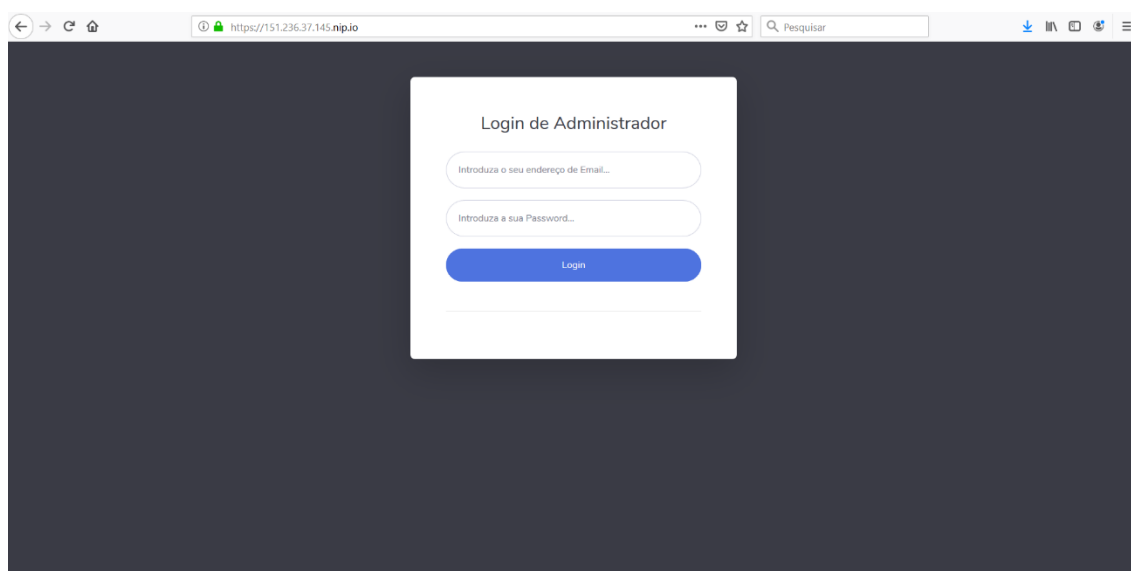


Figura 8: Página de Login.

Na página principal (Figura 9) são mostrados todos os registos na base de dados temporária *temp_civil_protection* em que a *flag* editado se encontre a *false*. São dadas duas opções ao administrador de sistema para cada registo: editar ou eliminar. Se este clicar no botão Editar, terá a possibilidade de normalizar os dados relacionados com a via encerrada em questão. Após editar/adicionar os dados em questão este poderá guardar as alterações, o que fará com que este registo seja adicionado na tabela principal *civil_protection*

²⁴ <https://en.wikipedia.org/wiki/Bcrypt>

e que a *flag* editado do registo da tabela `temp_civil_protection` seja colocada a *true*.

Ações	Nome	Localização	Estado	Data de Encerramento	Data de Reabertura
Editar Eliminar	Estrada da Banda de Além	Entre a entrada para a Estrada do Pomar da Rocha e a entrada para a Estrada do Plco Ferreiro	Interrupção de Trânsito Automóvel	vazio	2017/05/08
Editar Eliminar	Editat N° 544/2018 - Caminho do Lombo dos Aguiar	Troço entre o Caminho Dr. Padre Abel Augusto da Silva e o Caminho do Laranjal	Interrupção	05/08/2019	05/11/2018
Editar Eliminar	Editat N° 21/2019 - Estrada Comandante Camacho de Freitas	Troço entre a estrada da Fundosa e o Caminho dos Saltos	Interrupção/Encerramento	vazio	21/01/2019

Figura 9: Página Principal.

Também existe a possibilidade de eliminar qualquer um dos registos, caso estes não sejam relacionados/importantes para a plataforma. Neste caso, o administrador de sistema, ao clicar no botão Eliminar irá somente atualizar a *flag* editado do registo da tabela `temp_civil_protection` a *true*, sem que seja realizada nenhuma nova entrada na tabela `civil_protection`.

Optou-se por utilizar uma *flag* para editar/eliminar os dados, pois deste modo nunca existe a perda de registos existentes na tabela `temp_civil_protection` devido a erro humano. A utilização desta *flag* e desta tabela temporária também permite, para além de informar que os dados já foram editados/eliminados, a verificação por parte do servidor Node.js²⁵ da existência ou inexistência dos dados relativos às vias encerradas. Deste modo evita-se também a duplicação dos dados, uma vez que perderíamos as referências caso estes fossem editados diretamente numa única tabela.

A normalização dos dados é feita de forma a que os dados possam ser utilizados na plataforma Android. Isto torna-se importante pois os dados ao serem recolhidos a partir do Serviço Regional de Proteção Civil da Madeira, não se encontram normalizados (p. ex. as datas não se encontram inseridas sempre do mesmo modo, por vezes no mesmo campo é inserida a data e as horas), possibilitando assim que os dados não venham a ser inseridos na base de dados da plataforma Android, ou até mesmo a despoletarem erros que possam impossibilitar o uso da aplicação.

²⁵ <https://nodejs.org/en/>

4.2 DNS

Ao aceder a um *website*, como por exemplo o da Google (<https://www.google.com>), este é traduzido para um endereço IP, neste caso o IP 216.58.211.46. O único problema destes domínios é que estes normalmente são pagos. No caso do site de administração, este pode ser acedido através do domínio <https://www.151.236.37.145.nip.io>, que embora possua o endereço IP incorporado no mesmo, é um nome de domínio que resolve para o endereço IP 151.236.37.145.

De modo a não haver a necessidade da compra de um domínio e tendo em conta a diminuição de despesas, foi utilizado o serviço fornecido gratuitamente pela Exentrique Solutions²⁶, neste caso um servidor de DNS que permite o mapeamento de um *hostname* para um endereço IP utilizando para isso os DNS *wildcards*.

Este serviço tem como nome nip.io, correndo num servidor de DNS personalizado. Sempre que um cliente tente aceder a um domínio nip.io²⁷, o servidor de DNS extrai o endereço IP do domínio e envia-o na resposta. Essencialmente, é um serviço que utiliza habilmente o endereço IP como parte do *domain name* para retornar automaticamente o pedido de DNS para esse IP. Sempre que um novo pedido é feito, este pedido irá parar aos servidores de DNS do nip.io, sendo posteriormente reencaminhado para o servidor da plataforma.

A grande razão da utilização deste serviço deveu-se à implementação de HTTPS tanto no servidor HTTP Apache como no servidor Node.js. Para ser possível a utilização dos certificados SSL (estes certificados têm a validade de 90 dias), fornecidos gratuitamente pela autoridade de certificação Let's Encrypt²⁸, tornou-se necessária a criação de um *domain name*, uma vez que esta entidade de certificação não permitia a criação de certificados para endereços IP privados ou públicos. Embora este não seja no fundo um *domain name*, uma vez que apenas utiliza o IP público como *second-level domain* e o nip.io como *top-level domain*, este permitiu que fosse possível a utilização dos certificados SSL sem nenhum custo.

Uma vez que a implementação do atributo de segurança era de extrema importância para a plataforma em questão, e de modo a evitar gastos desnecessários, foi utilizado este serviço, que posteriormente

²⁶ <http://exentriquesolutions.com/>

²⁷ <https://nip.io/>

²⁸ <https://letsencrypt.org/>

permitiria a utilização de certificados SSL. Embora a validade dos certificados seja de menor duração que outros certificados encontrados no mercado, o Let's Encrypt recaiu como principal escolha por ser gratuito e de fácil implementação.

4.3 HTTPS

O protocolo HTTPS é utilizado para que exista uma comunicação segura entre os diversos componentes de uma rede, sendo neste caso em questão entre o servidor e a aplicação Android e entre o servidor e a página de administração. Deste modo é possível implementar o atributo segurança, garantindo a privacidade e a segurança dos dados enviados entre o servidor e os clientes e vice-versa. Este protocolo permite também que as credenciais introduzidas pelo administrador de sistema na página de administração sejam enviadas para o servidor da forma mais segura possível.

Para ativar o SSL no servidor foram primeiramente utilizados os certificados *self-signed* que são criados e geridos pelo próprio administrador de sistema. Para a criação destes certificados foi utilizada a biblioteca de *software* OpenSSL²⁹, que contém uma implementação *open-source* dos protocolos SSL e TLS. Embora a implementação de certificados *self-signed* seja gratuita, esta apresenta alguns problemas. Primeiro, estas não utilizam uma CA (*Certificate Authority*), o que faz com que os *browsers* e outras plataformas não confiem nestas certificações. Outro dos problemas tem a ver com a usabilidade, pois quando os clientes tentarem validar os certificados fornecidos, será gerada uma mensagem de erro indicando que o *browser* não confia nos certificados gerados. Isto significa, que se alguém tentar aceder ao *website*, o *browser* irá avisar que a validade e os certificados de segurança não conseguiram ser determinados. Outras plataformas podem mesmo não se conectar ao servidor, pois não os consideram certificados válidos. Para superar este erro seria necessária a instalação do certificado em cada cliente. Estes certificados são geralmente utilizados para testes e raramente são utilizados em ambiente de produção. Embora os problemas acima descritos tenham sido superados com a instalação local dos certificados nos *browsers* e posteriormente na plataforma Android, foi decidido seguir outra abordagem, que evitaria problemas futuros.

Numa segunda e última abordagem foi utilizada a autoridade de certificação Let's Encrypt, que é uma entidade sem fins lucrativos gerida

²⁹ <https://www.openssl.org/>

pelo ISRG, fornecendo certificados X.509 para encriptação TLS sem nenhum custo. O Let's Encrypt fornece um agente (Certbot³⁰) que utiliza o protocolo ACME para primeiramente validar o domínio do servidor *web* e, posteriormente para criar as chaves privadas e os *requests* CSR para a criação de certificados válidos.

Na Figura 10 é representado o funcionamento do servidor que, caso seja utilizado o protocolo HTTP num *request*, este será redirecionado passando assim a utilizar o protocolo HTTPS. Outro dos pontos fulcrais deste redireccionamento reside no URL do *request*, o que ditará para onde o pedido será redirecionado (servidor HTTP Apache ou servidor Node.js). A base de dados é partilhada tanto pela página de administrador que corre no servidor HTTP Apache como pelo servidor Node.js. O URL para aceder a ambos os servidores é o mesmo, apenas sendo diferenciado no *path* sendo passado. Se o URL possuir um *file path* como por exemplo *index.php*, este será reencaminhado para o servidor HTTP Apache, caso o URL apenas possua um *path*, como por exemplo *getCivilProtection*, este será reencaminhado para o servidor Node.js. No fundo é aproveitado o mesmo URL para aceder à página de administração e também para serem executadas as REST APIs no servidor. Quanto à disponibilidade destes serviços tanto as REST APIs como a página de administrador não apresentaram quaisquer problemas ou erros, uma vez que foi sempre possível aceder a estes serviços. Também, a utilização do *package* *forever*³¹ no servidor Node.js permite, caso o servidor pare devido a algum erro ou exceção, que este seja reiniciado automaticamente, enviando um email para o administrador de sistema.

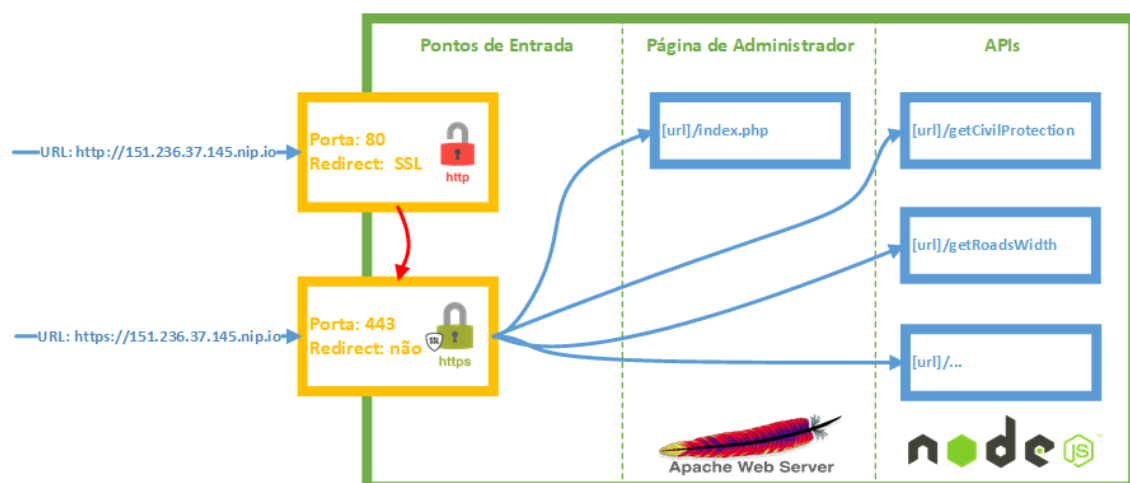


Figura 10: Redireccionamento dos requests.

³⁰ <https://certbot.eff.org/>

³¹ <https://www.npmjs.com/package/forever>

Na Figura 11 é representada uma análise à configuração do servidor SSL, utilizando para isso o serviço *online* gratuito SSL Labs³², fornecido pela Qualys. O *rating* que foi obtido pelo teste relativamente à segurança das configurações SSL implementadas no servidor, foi a nota A, sendo que a nota mais alta fornecida pelo site é A+.

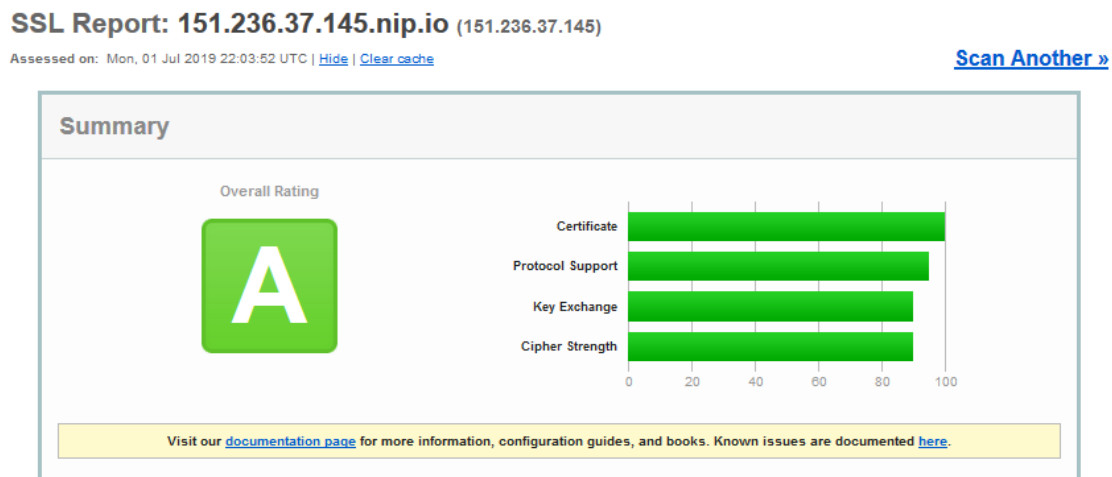


Figura 11: Análise da configuração do servidor web SSL utilizando o SSL Labs.

A metodologia utilizada para a atribuição destas notas baseia-se em 4 etapas:

1. É verificado o certificado de modo a averiguar se este é válido e confiável.
2. É inspecionada a configuração do servidor, baseando-se em 3 categorias:
 - a. Protocolo de suporte;
 - b. Suporte à troca de chaves (*key exchange support*);
 - c. Suporte de cifras (*cipher support*).
3. São combinadas as pontuações de cada categoria para uma pontuação geral (expressa como um número entre 0 e 100). Um 0 em qualquer uma das categorias irá influenciar, e muito, na pontuação geral, uma vez que representará que existe uma falha bastante importante no certificado. Por fim, é calculada a nota, tal como é representado na Tabela 5.
4. De seguida são aplicadas uma série de regras para lidar com alguns aspetos da configuração do servidor, que serão posteriormente expressas pelas notas de A até F. Algumas regras aumentam a nota para (A+), para recompensar algumas configurações que sejam excecionais. Em determinadas situações,

³² <https://www.ssllabs.com/ssltest/analyze.html?d=151.236.37.145.nip.io>

são evitadas as notas de A até F, quando são identificadas incompatibilidades no nome do certificado (nota M) ou quando o certificado do site não é confiável (nota T).

Pontuação Numérica	Nota
Pontuação ≥ 80	A
Pontuação ≥ 65	B
Pontuação ≥ 50	C
Pontuação ≥ 35	D
Pontuação ≥ 20	E
Pontuação < 20	F
Incompatibilidades no nome do certificado	M
Certificado não confiável	T

Tabela 5: Atribuição de notas utilizando o SSL Labs.

Quando não existe confiabilidade no certificado, o nível de segurança real não importa, uma vez que qualquer pessoa mal-intencionada poderá aproveitar-se desta fragilidade. Um certificado que não seja confiável (ou seja, não sendo assinado por uma autoridade de certificação conhecida) falha ao impedir ataques do tipo *man-in-the middle* e torna o SSL efetivamente inútil. Um certificado que esteja incorreto por qualquer outra razão (p. ex. um certificado que tenha expirado) destrói a confiança, pondo em risco a segurança de toda a plataforma. Por estes motivos, qualquer um dos seguintes problemas resulta imediatamente numa pontuação de 0: certificados ainda não válidos, certificados expirados, certificados *self-signed*, utilizar um certificado que não seja confiável (autoridade de certificação desconhecida ou algum outro erro de validação), assinaturas de certificados inseguras (MD2 ou MD5) ou até mesmo chaves inseguras.

Na Figura 12 mostra-se o resultado do mesmo teste, mas utilizando outra plataforma, o SSL Checker³³, fornecido pelo SSL Shopper. Este permite diagnosticar problemas com as instalações de certificados SSL, verificando se estes se encontram corretamente instalados, se são válidos, confiáveis e se não irão apresentar nenhum erro aos utilizadores.

Este teste já havia sido feito quando estávamos utilizando os certificados *self-signed* e a nota dada havia sido a mesma apenas falhando nalguns parâmetros relacionados com a entidade que atribuía os certificados, que nesse caso em questão era inexistente.

³³ <https://www.sslshopper.com/ssl-checker.html#hostname=151.236.37.145.nip.io>

Server Hostname

151.236.37.145.nip.io
Check SSL

✓

151.236.37.145.nip.io resolves to 151.236.37.145

✓

Server Type: Apache/2.4.27 (Ubuntu)

✓

The certificate should be trusted by all major web browsers (all the correct intermediate certificates are installed).

✓

The certificate was issued by Let's Encrypt.
Write review of Let's Encrypt


✓

The certificate will expire in 77 days.
Remind me

✓


The hostname (151.236.37.145.nip.io) is correctly listed in the certificate.

Server



Common name: www.151.236.37.145.nip.io
SANs: 151.236.37.145.nip.io, www.151.236.37.145.nip.io
Valid from June 19, 2019 to September 17, 2019
Serial Number: 035a543e33ecf847570b2927c565c737bfac
Signature Algorithm: sha256WithRSAEncryption
Issuer: Let's Encrypt Authority X3

Chain



Common name: Let's Encrypt Authority X3
Organization: Let's Encrypt
Location: US
Valid from March 17, 2016 to March 17, 2021
Serial Number: 0a0141420000015385736a0b85eca708
Signature Algorithm: sha256WithRSAEncryption
Issuer: DST Root CA X3

Figura 12: Análise da configuração do servidor web SSL utilizando o SSL Checker.

Por fim, pode-se afirmar, tendo em conta os resultados obtidos por ambos os testes, que os certificados instalados no servidor e fornecidos pela autoridade de certificação Let's Encrypt, são certificados de qualidade e que permitem proteger a plataforma como um todo, fornecendo a segurança dos dados quando estes tenham de ser transferidos entre os vários componentes do sistema. O único problema a identificar, como referido anteriormente, é a validade de 90 dias destes certificados, mas tendo em conta que estes são gratuitos, é uma boa solução quer para esta ou para outras plataformas.

4.4 Servidor

O servidor foi desenvolvido de modo a fornecer e a receber dados de dois componentes, a página de administração e a aplicação Android. Também são recebidos dados a partir de fontes externas à plataforma, dados

que são de extrema importância para o funcionamento de todo o sistema. No servidor são utilizadas duas tecnologias: o servidor HTTP Apache e o servidor Node.js. Foram utilizadas estas duas tecnologias de modo a existir uma separação de responsabilidades, sendo que o servidor HTTP Apache ficou responsável pela página de administração e o servidor Node.js responsável pela centralização e tratamento dos dados provenientes de várias fontes e também pelo fornecimento de REST APIs, que poderão ser utilizadas pela aplicação Android para realizar pedidos sobre os dados obtidos anteriormente.

Como basicamente o servidor HTTP Apache já foi explicado anteriormente, nesta secção apenas será explicado o funcionamento do servidor Node.js. O Node.js pode ser definido como um ambiente de execução JavaScript *server-side*. Algumas das vantagens em se utilizar um servidor em Node.js prende-se com o facto de este ser leve e multiplataforma, permitindo uma adaptação rápida. Outra das vantagens tem a ver com a quantidade de *packages* existentes que pretendem solucionar determinados problemas. Se precisarmos de uma solução para um problema é bem provável que já alguém tenha tido esse problema e criado um *package* que poderá ser utilizado por toda a comunidade. Normalmente estes *packages* são *open-source* e é possível a inspeção ou modificação do código no GitHub. O Node.js por ser multiplataforma permite também a sua implementação em sistemas Linux, o que permite reduzir substancialmente os custos de manter um servidor *online*.

Abaixo serão explicados os diferentes módulos que compõem o servidor Node.js e as suas respectivas funcionalidades. Cada módulo foi preferencialmente construído sobre as funcionalidades de *packages* específicos, de modo a que cada módulo fosse apenas responsável por determinadas funcionalidades específicas. O único módulo em que isto não aconteceu foi o módulo *civil_protection*, que devido à sua complexidade, e devido a apenas realizar um conjunto de instruções relacionadas com a obtenção de dados provenientes do Serviço Regional de Proteção Civil da Madeira, possui diversos *packages* dentro do mesmo módulo.

4.4.1 Módulo Civil Protection

Este módulo é onde é realizada toda a lógica relacionada com a obtenção dos dados provenientes do *website* do Serviço Regional de Proteção Civil da Madeira. Este módulo é executado diariamente às 19 horas,

sendo para isso utilizado o *package* *cron*³⁴, que permite o agendamento de tarefas (*jobs*) despoletando um *trigger* sempre que os valores de data e tempo introduzidos sejam satisfeitos. Caso seja necessária a repetição das tarefas de forma diária, os valores do dia e do mês são passados a * e o valor do dia da semana é passado de 0-6, em que 0 é o Domingo e 6 é o Sábado. Deste modo seria possível também apenas correr as tarefas em dias de trabalho (1-5).

Os dados são basicamente obtidos todos os dias à mesma hora sendo que, caso os dados não estejam normalizados, seja enviado um email para o administrador de sistema informando que existem registos que precisam de ser normalizados. Não são enviadas notificações aos utilizadores a esta hora, uma vez que os encerramentos de vias poderão acontecer numa data muito posterior à data de emissão dos editais.

O módulo *cron* também serve para o envio de notificações para os dispositivos móveis sempre que exista uma estrada encerrada no dia seguinte. Isto serve para que os utilizadores tomem conhecimento de antemão que uma via irá estar encerrada no dia seguinte. O que acontece é que a primeira vez que uma estrada encerrada é adicionada pelo Serviço de Proteção Civil corresponde ao dia em que o respetivo edital é emitido, não existindo uma data para a sua emissão, podendo ser com um, dois ou mais meses de antecedência, podendo assim ser esquecida a informação pelo utilizador. Para isso foi implementado o envio de uma notificação no dia anterior ao encerramento de uma via, visando assim lembrar ao utilizador o encerramento dessa mesma via.

Neste módulo é realizado o *download* do KML a partir do *website* do Serviço Regional de Proteção Civil da Madeira. É então realizado o *parsing* destes dados para *JSON*, não porque não fosse possível a sua leitura realizando o *parsing* através de algum *parser* XML (testado com o *package* *xml-js*³⁵), mas devido a padronizar os *requests* e respetivos resultados.

Para cada via encerrada retornada é verificado se esta contém algum campo vazio. Caso possua algum campo vazio, este campo é preenchido com a *string* “vazio”. Posteriormente é também verificado se esta já se encontra introduzida na tabela *temp_civil_protection*, de modo a não existir duplicação de dados. Caso já exista na base de dados, os dados para esta via encerrada são ignorados. Caso ainda não exista, é realizado um *request* ao serviço do Overpass API, que permite a obtenção do nome da estrada, para isso apenas bastando passar as coordenadas obtidas a partir do *website* do

³⁴ <https://www.npmjs.com/package/cron>

³⁵ <https://www.npmjs.com/package/xml-js>

Serviço Regional de Proteção Civil da Madeira e também o valor do raio (*around*) de forma a limitar o tamanho da zona onde será realizada a pesquisa por uma determinada estrada. Por defeito, o valor do raio é colocado a 20 (metros).

Caso o *request* anterior retorne dados para mais do que uma via, são efetuados, de forma recursiva, *requests* com as mesmas coordenadas, apenas diminuindo em uma unidade o valor do raio. Caso o *request* não retorne nenhuma via, os dados da via encerrada são ignorados, não sendo guardados na base de dados. Esta situação acontece raramente, mas foi algo que teve de ser tratado, uma vez que nem sempre as informações fornecidas pelo Serviço Regional de Proteção Civil da Madeira são referentes ao encerramento de estradas.

Com o nome da via obtido no *request* anterior, é realizado o *request* seguinte, também utilizando o serviço do Overpass API, passando o nome da via como parâmetro. Este permite a obtenção de todas as coordenadas que pertencem a uma determinada via. De salientar que o Overpass API utiliza os dados do OSM³⁷, em que cada via se divide em vários *way(s)* que por sua vez possuem um conjunto de *nodes* (estes últimos possuem as referências às coordenadas de cada nó da via).

Através das coordenadas de início e de fim de cada *way*, são obtidos os LinkIds utilizando para isso o serviço Directions API da MapQuest (Find Link Id – GET³⁸), que serão utilizados posteriormente na aplicação móvel. Os LinkIds tornam possível a apresentação das vias encerradas na aplicação móvel e também permitem que ao serem passados como valores no parâmetro *mustAvoidLinkIds* do *request* das rotas de navegação, que as rotas sejam traçadas evitando neste caso as vias que se encontram encerradas. O *request* para traçar as rotas de navegação também utiliza o serviço Directions API da MapQuest (Route – GET³⁹).

Por fim é verificado se os dados para cada via encerrada já se encontram normalizados ou não (esta normalização é obrigatória principalmente no caso das datas de encerramento e de reabertura, pois nem sempre são introduzidas e quando introduzidas podem possuir diversos formatos). Muitas das vezes os dados necessitam desta normalização, especialmente no caso das datas, que se encontram nos PDFs dos editais emitidos pelas

³⁷ <https://www.openstreetmap.org>

³⁸ <https://developer.mapquest.com/documentation/directions-api/find-link-id/get/>

³⁹ <https://developer.mapquest.com/documentation/directions-api/route/get/>

entidades municipais, mas que não se encontram no *website* do Serviço Regional de Proteção Civil da Madeira.

Caso já se encontrem normalizados, os dados são guardados nas tabelas `civil_protection` e `temp_civil_protection`, sendo que nesta última a *flag* editado é colocada a *true*. Caso os dados ainda não se encontrem normalizados, necessitando para isso de intervenção humana, estes são guardados na tabela `temp_civil_protection` com a *flag* editado a *false*.

De salientar que após cada *request* realizado ao Overpass API há um compasso de espera de 5 segundos, antes de realizar um novo *request*. Isto deve-se, primeiramente à limitação de um *request* por segundo imposta pelo OSM, que o Overpass API tem como base. Para além disso, após cada *request*, o servidor do Overpass API adiciona um tempo extra antes de permitir que seja feito um novo *request* a partir do mesmo endereço IP. A duração do tempo extra é proporcional ao tempo de execução da *query* (nos *requests* ao Overpass API, a linguagem utilizada baseia-se no Overpass QL⁴⁰, permitindo realizar *queries* à base de dados do OSM). O fator utilizado é maior quanto maior for a carga no servidor. Caso contrário, existiria a possibilidade de bloquear uma CPU inteira do servidor para um único cliente. Isto seria injusto para outros utilizadores, e por isso este fator depende do número de utilizadores a aceder ao servidor de forma simultânea.

Um segundo mecanismo faz com que cada servidor possa atrasar a resposta a um *request* de até 15 segundos, caso existam demasiados pedidos a serem efetuados em simultâneo. Por esta razão o Overpass API utiliza um sistema de quotas por endereço IP, que apenas são contabilizadas tendo em conta os tempos de execução após os atrasos. Este mecanismo serve para facilitar o envio de vários *requests* de forma assíncrona. No entanto, se a carga no servidor for razoável, não é necessária a implementação da gestão de quotas no lado do cliente.

4.4.2 Módulo `livesql`

Este módulo, como explicado anteriormente, baseia-se no *package* `live-sql`⁴¹, funcionando como um *replication slave* que fica à escuta de modificações numa tabela ou até mesmo em toda a base de dados. Este *package* pode identificar que tipo de comandos DML foram executados (permitindo filtrar pelos mesmos, quer seja um `INSERT`, um `UPDATE` ou

⁴⁰ https://wiki.openstreetmap.org/wiki/Overpass_API/Overpass_QL

⁴¹ <https://www.npmjs.com/package/live-sql>

um DELETE). Permite também apenas filtrar pelas tabelas desejadas, sendo que neste caso apenas está sendo utilizada a tabela *civil_protection*. Outra das opções existentes tem a ver com modificações a nível do *schema* da base de dados, como por exemplo inserir, atualizar ou eliminar colunas ou os tipos de dados das colunas das tabelas, inserir, atualizar ou eliminar tabelas na base de dados... No entanto, devido às bases de dados utilizadas serem maioritariamente estáticas, sem grandes mudanças a nível do *schema*, esta opção embora testada, não chegou a ser utilizada.

Embora este módulo tenha sido construído utilizando para isso o *package live-sql*, este é completamente dependente dos *binary logs* gerados a partir da base de dados MySQL. Sempre que um comando DML seja executado na base de dados este é posteriormente guardado nos *binary logs*. Devido a esta extrema dependência será então primeiro explicada a função dos *binary logs* dentro do sistema e então a forma como o *package live-sql* se aproveita destes para detetar mudanças na base de dados e posteriormente, com a ajuda de outros *packages*, atualizar as bases de dados dos dispositivos móveis.

4.4.2.1 Binary Logs

Os *binary logs* contêm eventos que descrevem as alterações efetuadas numa base de dados, tais como alterações no *schema* das tabelas ou nos registos lá existentes. Os *binary logs* contêm também informações sobre o tempo de execução de cada comando. Estes *logs* têm duas finalidades importantes:

- Para a replicação de dados, em que as alterações feitas num servidor principal (*master*) são guardadas num *binary log* que irá fornecer um registo das alterações de dados a serem enviadas aos servidores secundários (*slave*).
- Certas operações de recuperação de dados requerem a utilização de um *binary log*. Ao realizar a recuperação de dados numa base de dados, utilizando os *backups* já existentes, é possível atualizar os dados que não foram guardados desde o último *backup*, utilizando para isso os *binary logs* como complemento dos *backups*, para restaurar uma base de dados com os dados mais antigos através dos *backups* e com os dados mais recentes através dos *binary logs*.

Os *binary logs* não são utilizados para guardar comandos como SELECT ou SHOW que não modificam os dados na base de dados. Apenas os comandos DML que modifiquem a base de dados, tais como INSERT,

UPDATE, DELETE ou comandos que modifiquem o *schema* da base de dados são tidos em conta.

Para monitorizar os ficheiros de *log* que já foram criados, existe um ficheiro onde são guardados os nomes de todos os *binary logs* já criados. Por padrão, este tem o mesmo nome de base dos *binary logs*, apenas diferindo na sua extensão `‘.index’` (Figura 13).

A criação de diversos ficheiros de *log* tem em conta as configurações inicialmente definidas para os *binlog files*, em que parâmetros tais como a localização dos ficheiros a serem criados, o seu tamanho (100MB) e a validade destes (10 dias) irão influenciar a quantidade de ficheiros a serem criados. Neste caso quanto menor o tamanho dos ficheiros e quanto mais a base de dados for utilizada maior será a quantidade de *binlog files*. A escolha de um tamanho nem muito grande nem muito pequeno ajudará na consulta aos dados que foram alterados, pois um ficheiro muito grande demorará muito tempo a ler e um ficheiro muito pequeno levará a que devido à indexação destes ficheiros, que tenhamos de aceder a mais do que um ficheiro para obter as últimas alterações aos dados da base de dados.

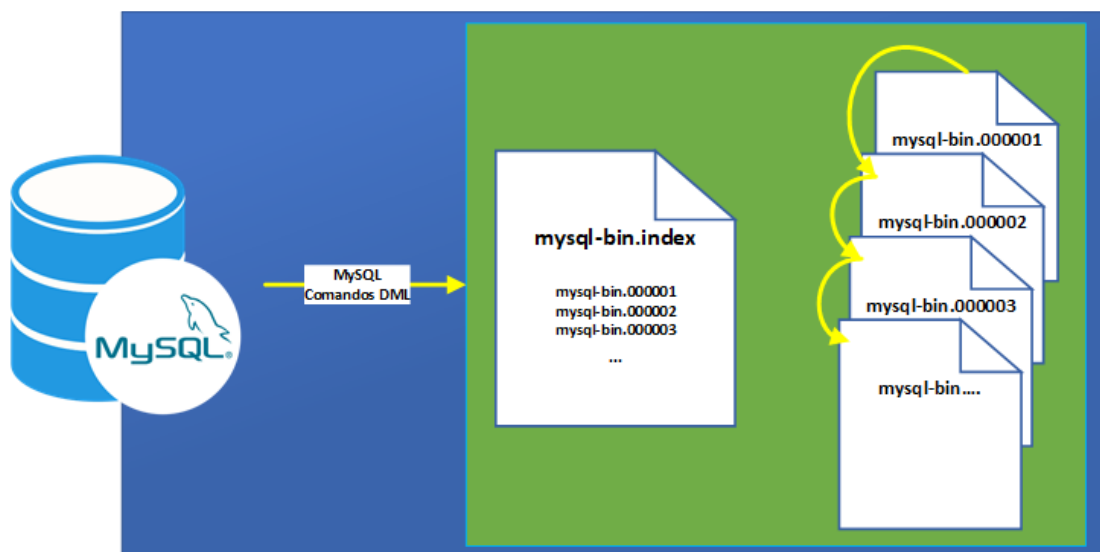


Figura 13: Esquema do funcionamento dos binlogs nas bases de dados MySQL.

4.4.2.2. Live-sql package

O *package* live-sql utilizado no servidor Node.js é um *binlog listener* que fica à escuta de modificações nos ficheiros *binlog*. Sempre que são introduzidas novas modificações nos ficheiros de *log* é despoletado um *trigger* que permite que posteriormente seja feito o *broadcast* para os dispositivos Android dos comandos DML introduzidos anteriormente na base de dados pelo módulo *civil_protection*. Neste caso em questão apenas são filtrados os comandos INSERT, UPDATE e DELETE, sendo ignorados

todos os comandos que impliquem modificações ao *schema* da base de dados. Após ser feito o *broadcast* destes comandos, a aplicação realiza cada comando recebido nas suas bases de dados locais (dispositivos Android).

O *broadcast* é realizado através de *push notifications*, sendo estas notificações enviadas para todos os dispositivos Android, utilizando para isso a biblioteca Socket.io. Caso os utilizadores não se encontrem conectados à Internet, o envio destes mesmos dados é realizado posteriormente, utilizando para isso os *timestamps* de acesso ao servidor de cada cliente. É realizada uma verificação dos *timestamps* de modificação dos dados da tabela *civil_protection* e caso o último *timestamp* de acesso ao servidor por parte do cliente seja anterior ao *timestamp* de alguma modificação nalgum registo na tabela *civil_protection*, os comandos DML executados são enviados, juntamente com os dados modificados, para o dispositivo Android do respetivo cliente.

De salientar que apenas os registos inseridos na tabela *civil_protection* despoletam este *trigger*, uma vez que já se encontram normalizados e prontos a serem inseridos nas bases de dados da aplicação Android.

4.4.3. Módulo nodemailer (*nodemailer package*)

O *package* nodemailer⁴³ é *open-source* e torna possível o envio de *emails* a partir do servidor Node.js, baseando-se para isso no protocolo SMTP. Para além de ser possível adicionar texto e HMTL, o nodemailer permite também o envio de anexos. O nodemailer necessita também de um serviço pelo qual seja possível o envio de *emails*, tendo sido utilizado para esse propósito o serviço do Gmail.

Embora o Gmail tenha sido o serviço utilizado, por ser uma das formas mais rápidas e simples de enviar *emails*, este serviço espera que a plataforma que está enviando os *emails* seja um utilizador real e não um *robot* ou um sistema automático de envio de *emails*. Por esta razão, são executados muitos métodos heurísticos para cada tentativa de *login*, bloqueando atividades suspeitas de forma a proteger os utilizadores de tentativas de invasão das suas contas.

Isto faz com que o Gmail acabe identificando a plataforma como uma aplicação “menos segura”, existindo uma necessidade inicial de o administrador de sistema cujas credenciais (*email* e *password*) estão sendo

⁴³ <https://nodemailer.com/about/>

utilizadas para o envio dos *emails* a partir da plataforma, de configurar a sua conta Gmail, de modo a permitir a utilização das suas credenciais em aplicações que são consideradas “menos seguras” (Figura 14).

← Acesso a aplicações menos seguras

Algumas aplicações e dispositivos utilizam tecnologia de início de sessão menos segura, o que deixa a sua conta vulnerável. Recomendamos que **desative** o acesso para estas aplicações. Se pretender utilizar estas aplicações apesar dos riscos, **ative** o acesso. [Saiba mais](#)

Permitir aplicações menos seguras: ATIVADO 

Figura 14: Formulário de ativação de aplicações menos seguras.

Cada vez que um novo registo é obtido a partir do Serviço Regional de Proteção Civil da Madeira, e caso os dados desse registo não se encontrem normalizados, é enviado um *email* para o administrador de sistema de modo a informar que existem dados que necessitam de ser normalizados.

Novo registo inserido na tabela temp_civil_protection:

ID: 341

NAME: Edital N.º 299/2019 - Rua da Carreira

LOCALIZAÇÃO: Rua da Carreira; Rua das Pretas; Rua da Mouraria

ESTADO: Interrupção

JUSTIFICAÇÃO: Procissão

RESPONSABILIDADE: CMF

EDITAL/DOCUMENTO: <https://drive.google.com/open?id=1rACRYwTa7ftqChgQE6scNEqP5EyMKv>

COORDENADAS: 32.6497233, -16.9118504

DATA REABERTURA: 29/06/2019

DATA ENCERRAMENTO: 29/06/2019 - 18h30



Figura 15: Email informativo da existência de um novo registo na tabela temp_civil_protection.

Neste *email* são enviados, de forma informativa, os dados obtidos a partir do referido *website*, sendo também enviado, caso exista, o PDF com o edital para a via encerrada correspondente. Isto permite que o administrador de sistema, caso necessite, possa aceder à informação do edital, de modo a complementar alguma informação que se encontre em falta (Figura 15).

Para além disso, este módulo serve também para o envio de notificações relativas a algum erro que tenha ocorrido no servidor Node.js, bem como alguma reativação que tenha sido feita automaticamente pelo *package forever* (Figura 16).

Alguns dos dados enviados na notificação de erro são:

- O *timestamp*, permitindo assim ao administrador de sistema, conhecer a altura em que ocorreu o erro;
- No caso de ocorrer algum erro durante um *request*, este indica qual o URL do *request*, de modo a conhecer quais os dados que foram retornados em determinado pedido;
- Os erros, caso exista algum *log* de erro, este será enviado na notificação ao administrador de sistema.

DATE AND TIME: Mon Jul 01 2019 12:36:06 GMT+0100 (BST)

REQUEST URL: [https://www.overpass-api.de/api/interpreter?data=\[out.json\]\[timeout:25\].area\[name='Ilha da Madeira'\].way\[name='Rua de São Pedro'\]\(area\);\(._>\).out body;](https://www.overpass-api.de/api/interpreter?data=[out.json][timeout:25].area[name='Ilha da Madeira'].way[name='Rua de São Pedro'](area);(._>).out body;)

MISSING KEY: tags

Figura 16: Exemplo de uma notificação de erro.

No fundo, este módulo, embora não muito importante em termos de funcionalidades do sistema, é um elemento fulcral de modo a que o administrador de sistema consiga facilmente encontrar ou tomar conhecimento de novos registos na base de dados que necessitem de normalização. Caso os dados já estejam normalizados nenhuma notificação é enviada, permitindo assim que o administrador de sistema se possa focar nos registos que realmente precisem de atenção.

Por outro lado, e não menos importante, é o envio de notificações de erro no servidor e respetivo envio dos ficheiros de *log*. Este mecanismo permite ao administrador de sistema identificar possíveis falhas no servidor, bem como facilitar o seu trabalho, pois através dos *logs* de erro conseguirá mais facilmente identificar o problema e respetivas soluções. Deste modo, os erros ao serem diretamente reportados para a caixa de correio do administrador de sistema, permite que este possa ter conhecimento do problema de forma praticamente espontânea, o que faz com que este possa

realizar *hotfixes* no caso de erros mais graves e que impossibilitem o funcionamento normal do servidor ou *bugfixes* quando estes erros sejam de menor prioridade. No fundo, poder informar de forma mais célere irá fazer com que a resolução desses mesmos problemas possa ser mais rápida, mantendo o servidor em *uptime* o maior tempo possível.

4.4.4. Módulo `socket_io` (*socket.io package*)

O *package* Socket.io é uma biblioteca JavaScript que permite a comunicação em tempo real, bidirecional e baseada em eventos entre o cliente (*browser*, aplicação...) e o servidor. Deste modo este *package* permite que seja feito um *broadcast* dos dados provenientes do *website* do Serviço Regional de Proteção Civil da Madeira sempre que estes se encontrem normalizados.

Sempre que é introduzido um novo registo na tabela `civil_protection` da base de dados, é realizado o *broadcast* desse registo para todos os dispositivos Android que se encontrem ligados à *Internet* nesse momento. Posteriormente, a aplicação Android irá inserir estes novos dados nas bases de dados locais, permitindo assim que esta informação seja mostrada ao utilizador através do mapa da aplicação.

Sempre que um dispositivo Android aceda ao servidor, é criado um registo desse dispositivo na tabela `android_device` com os dados que identifiquem esse dispositivo Android e respetivo *timestamp* de acesso. É verificado se existem novos registos ou alterações nos registos já existentes na tabela `civil_protection`. Caso existam novos dados, com um *timestamp* posterior ao último *timestamp* de acesso do dispositivo ao servidor, é realizado o envio desses mesmos dados apenas para esse dispositivo Android.

Caso o dispositivo já exista na tabela da base de dados apenas é atualizado o *timestamp* de acesso, permitindo deste modo que da próxima vez que este aceda ao servidor, seja possível conhecer quais as modificações que foram feitas na tabela `civil_protection` e que ainda não existem na base de dados local desse dispositivo.

Este módulo age conjuntamente com o módulo `livesql`, em que o `livesql` permite a identificação das modificações realizadas na tabela `civil_protection`, tanto de forma espontânea, em que o Socket.io realiza o *broadcast* dos dados, como quando existe a necessidade de atualizar os dados de um dispositivo móvel que não estivesse ligado à *Internet* no momento do

broadcast. Neste último, existe um mecanismo que, como explicado anteriormente, permite saber o último acesso do cliente ao servidor, obtendo assim apenas os dados posteriores ao *timestamp* do último acesso.

Deste modo é garantido que todos os dispositivos tenham os dados mais atualizados nas suas bases de dados, garantindo que a mesma informação seja mostrada em todos os dispositivos móveis que possuam esta aplicação Android instalada.

4.4.5. Módulo https (*https package*)

O *package* *https*⁴⁵, é utilizado conjuntamente com o *package* *express*⁴⁶ para gerar as APIs REST necessárias para a utilização na plataforma Android. As APIs permitem que, neste caso a aplicação Android, emita *requests* para solicitar um conjunto de informações (dados relativos à largura das vias rodoviárias).

Este módulo filtra, caso o URL possua como *path* *getRoadsWidth*, os *requests* relacionados com a largura das estradas. Estes *requests* são realizados pela aplicação Android, apenas devolvendo as estradas que tenham no máximo 3 metros de largura e que se encontrem no trajeto traçado pelo utilizador no mapa da aplicação. Este valor foi utilizado, tendo em conta a largura de uma viatura familiar.

Caso o URL possua como *path* *getCivilProtection*, este módulo irá filtrar os *requests* relacionados com os dados obtidos a partir do Serviço Regional de Proteção Civil da Madeira. Embora esta API não esteja sendo utilizada de momento, esta foi criada de modo a retornar todas as estradas que estejam ou venham a se encontrar encerradas. Esta API permitiu de início obter os dados para cada aplicação Android, sendo posteriormente substituída pelo mecanismo implementado no módulo *socket_io*. Por fim, esta permitiu conferir que os dados passados pelo módulo *socket_io* estavam corretos, verificando se os dados introduzidos na base de dados da aplicação Android eram os mesmos que os passados nesta API REST. Por outro lado, esta API permitiria também que qualquer cliente pudesse vir a consumir os dados, uma vez que estes retornavam todas as vias que estavam ou que estariam futuramente encerradas.

Foi utilizado o *package* *https* de forma a garantir a confidencialidade e segurança entre as comunicações estabelecidas entre o servidor e a

⁴⁵ <https://www.npmjs.com/package/https>

⁴⁶ <https://www.npmjs.com/package/express>

plataforma Android. Como foi descrito acima, através do Let's Encrypt, são obtidos neste caso, o certificado e a *key* que irão permitir o envio seguro das informações.

4.5 Base de Dados

A base de dados foi implementada utilizando o sistema de gestão de base de dados MySQL. Esta é uma base de dados relacional, tendo também a vantagem de ser *open-source*. Faz parte do *package* XAMPP, permitindo ser utilizada tanto pelo servidor Apache como pelo servidor Node.js.

Como referido anteriormente, a escolha de um sistema de gestão de base de dados relacional foi motivada por esta ser suportada pela maioria dos sistemas operativos, permitindo assim que seja possível mudar de plataforma de maneira mais fácil e segura. Outra das razões que levaram à implementação deste tipo de base de dados foi o armazenamento de dados que seriam externos à plataforma, e que normalmente seriam guardados como tipos de dados básicos. No entanto, após a aquisição de diferentes tipos de dados, nomeadamente dados geográficos ou espaciais, a possibilidade de utilizar bases de dados de grafos ou até mesmo orientadas a objetos teria um maior peso nessa decisão. Embora o MySQL tenha suporte a tipos de dados que possam ser guardados como dados espaciais (dados geográficos), as *queries* sobre relações entre tabelas podem levar muito tempo a serem processadas, principalmente quando a quantidade de dados armazenados aumenta. Isto ocorre porque as relações são implementadas utilizando para isso chaves estrangeiras (*foreign keys*), e que são consultadas ao realizar o JOIN dessas mesmas tabelas. A execução de JOINS pode ser demorado, especialmente se precisarmos de associar várias tabelas para executar consultas mais complexas.

As bases de dados de grafos armazenam as relações juntamente com os dados, permitindo que as relações entre os nós estejam fisicamente vinculadas na base de dados. O acesso a estas relações é tão imediato quanto o acesso a esses mesmos dados. Por outras palavras, em vez de existir a necessidade de realizar JOINS entre as tabelas da base de dados, as bases de dados de grafos permitem simplesmente obter as relações diretamente nas entradas da base de dados. Neste caso, as bases de dados de grafos não armazenam apenas as relações entre os diferentes objetos de forma nativa, permitindo a inclusão de diferentes tipos de objetos e diferentes tipos de relações no mesmo registo da base de dados. Por fim,

devido ao armazenamento de dados geográficos, as bases de dados de grafos apresentariam diversas vantagens comparativamente às bases de dados relacionais.

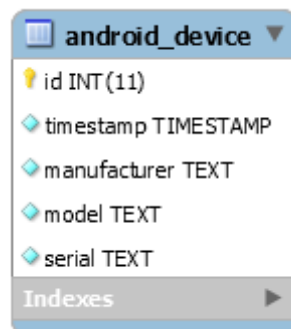
Para o armazenamento dos dados existem três bases de dados, cada uma delas com responsabilidades diferentes. Estas serão explicadas em seguida, permitindo conhecer as responsabilidades inerentes a cada uma delas, bem como os dados armazenados nas mesmas.

4.5.1 Base de Dados *closed_roads*

Esta base de dados é responsável pelo armazenamento de todos os dados provenientes do *website* do Serviço Regional de Proteção Civil da Madeira, bem como de todos os dados de geolocalização relacionados com as vias encerradas/condicionadas. Também são armazenados nesta base de dados os dados de acesso dos dispositivos móveis ao servidor, que posteriormente permitirão a atualização das bases de dados das aplicações caso os dispositivos móveis não se encontrem ligados à Internet no momento dos *broadcasts* contendo a alteração de alguma informação na tabela *civil_protection*.

4.5.1.1 Tabela *android_device*

A tabela *android_device* (Figura 17) armazena um conjunto de dados relacionados com os dispositivos Android que permite identificar cada dispositivo que aceda ao servidor. Deste modo, guardando os *timestamps* de acesso ao servidor, é possível saber que dados ainda não foram transmitidos para cada dispositivo Android, permitindo que apenas sejam enviados os dados que ainda não existam nesse mesmo dispositivo.



android_device	
id	INT(11)
timestamp	TIMESTAMP
manufacturer	TEXT
model	TEXT
serial	TEXT
Indexes	

Figura 17: Tabela *android_device*.

Normalmente o *broadcast* dos dados é feito para todos os dispositivos Android, mas caso estes não se encontrem ligados à *Internet*, não será possível a sua atualização num determinado momento. Por isso foi implementado este mecanismo de modo a que todos os dispositivos Android possam ter as suas bases de dados atualizadas, sendo assim uma replicação perfeita da tabela *civil_protection* na base de dados do servidor.

4.5.1.2 Tabela *temp_civil_protection*

A tabela *temp_civil_protection* (Figura 18) é onde são armazenados todos os dados provenientes do *website* do Serviço Regional de Proteção Civil da Madeira. Houve a necessidade de uma tabela temporária devido a que, frequentemente os dados não se encontrem normalizados, necessitando de uma normalização posterior por parte do administrador de sistema, de forma manual. Esta tabela possui também uma *flag* editado que permite à plataforma distinguir entre os dados que já foram normalizados e os que ainda necessitam de intervenção. Caso os dados não necessitem de qualquer normalização, no momento da obtenção dos dados estes já são armazenados nesta mesma tabela com a *flag* editado a *true*. Através desta *flag* é possível saber que dados necessitam de ser mostrados ao administrador de sistema na página de administração, e que ainda necessitem de normalização.

Outra das justificações para a utilização desta tabela, é a necessidade, no momento do *parsing* dos dados obtidos, de uma comparação entre os dados em bruto armazenados nesta tabela e os dados provenientes do *website* do Serviço Regional de Proteção Civil da Madeira. Como são comparados os dados, coluna a coluna, se estes fossem editados diretamente nesta tabela não existiria um *exact match* entre os dados da tabela e os dados do *website*. Neste caso, todas as edições de dados são armazenadas diretamente na tabela *civil_protection* que será explicada posteriormente.

4.5.1.3 Tabela *civil_protection_geodata*

A tabela *civil_protection_geodata* (Figura 18) é onde são armazenados os dados que não são obtidos a partir do *website* do Serviço Regional de Proteção Civil da Madeira, mas que são obtidos posteriormente, e que são diretamente relacionados com as vias encerradas obtidas, a partir das coordenadas fornecidas. Aqui são armazenados os *LinkIds* que serão passados como parâmetros nos *requests* de obtenção dos trajetos pretendidos pelos utilizadores e que permitirão que as rotas traçadas evitem as vias encerradas e/ou condicionadas. Devido a esta ser uma tabela definitiva, encontra-se diretamente ligada tanto à tabela *temp_civil_protection*, para

permitir verificar se os dados obtidos a partir do Overpass API e da Directions API da MapQuest já se encontram armazenados ou não, e com a tabela `civil_protection`, permitindo assim o *broadcast* destes dados para os dispositivos Android.

4.5.1.4 Tabela nodes

A tabela `nodes` (Figura 18) é onde são armazenadas todas as coordenadas que compõem as vias encerradas e que permitirão que, no fundo, estas possam ser mostradas no mapa da aplicação dos dispositivos Android. Esta encontra-se, tal como a tabela `civil_protection_geodata`, ligada diretamente com as tabelas `temp_civil_protection` e `civil_protection`.

4.5.1.5 Tabela civil_protection

A tabela `civil_protection` (Figura 18) é para onde são passados os dados que já foram normalizados pelo administrador de sistema, ou que já se encontrem normalizados no momento da obtenção dos dados pelo servidor Node.js.



Figura 18: Diagrama Entidade-Relação entre as quatro tabelas onde são guardados os dados obtidos a partir do Serviço Regional de Proteção Civil da Madeira.

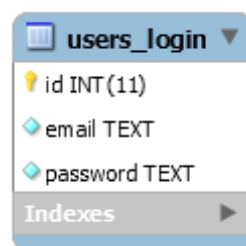
Esta tabela permitirá posteriormente que sejam enviados os dados para os dispositivos Android, dados que já se encontram normalizados, evitando assim que tenham de ser tratados antes da sua utilização dentro da plataforma. Esta tabela foi configurada para que fosse possível a criação de *binlogs*, que são atualizados a cada comando DML executado sobre a mesma. Assim, torna-se possível, a partir do *package* live-sql, a escuta de todas estas mudanças, tendo apenas que realizar o *listening* dos ficheiros de *log* no caso de alguma alteração.

4.5.2 Base de Dados users

Esta base de dados é responsável pelo armazenamento de todos os dados de *login* na página de administração. Foi utilizada esta abordagem de modo a evitar misturar dados sensíveis como são os *usernames* e as *passwords*, com outros dados que embora importantes, não possuam problemas relacionados com segurança. Deste modo foi procedida à respetiva separação de responsabilidades entres as várias bases de dados, fornecendo também uma maior segurança relativamente aos dados dos utilizadores.

4.5.2.1 Tabela users_login

A tabela *users_login* (Figura 19) é onde são guardados os dados de *login* da página de Administração, possuindo as credenciais (*email* e *password*) que serão posteriormente utilizadas para verificar se o *login* é válido ou não. No campo *password* é guardada a *password hash* que também permite uma maior segurança caso a base de dados seja vítima de um ataque.



users_login	
id	INT (11)
email	TEXT
password	TEXT
Indexes	

Figura 19: Tabela *users_login*.

4.5.3 Base de Dados roads_data

A base de dados *roads_data* possui todas as informações fornecidas pela Câmara Municipal do Funchal acerca da largura das estradas, possuindo

também informações obtidas a partir do serviço Nominatim ⁴⁷, disponibilizado pelo OSM. A necessidade de obter dados do Nominatim baseou-se na falta de dados de geolocalização no ficheiro Excel fornecido. Deste modo tornou-se necessária a obtenção de dados de geolocalização acerca das estradas estreitas, tendo para isso sido criadas *bounding boxes* que posteriormente poderiam facilitar a identificação dessas mesmas estradas.

4.5.3.1 Tabela funchal_roads

A tabela funchal_roads (Figura 20) possui a largura de todas as vias dentro do concelho do Funchal, bem como o tipo de pavimento, estado de conservação e respetivo nome da via. Esta tabela é utilizada para poderem ser mostradas na aplicação todas as vias que possuam uma largura que acabe dificultando o tráfego automóvel.

4.5.3.2 Tabela bounding_boxes

A tabela bounding_boxes (Figura 20) possui todas as *bounding boxes* correspondentes a cada via. Estas *bounding boxes* baseiam-se em dois pares de coordenadas. O primeiro par corresponde à coordenada mais a sudoeste e o segundo par à coordenada mais a nordeste. Cada via pode ter mais do que uma *bounding box*, dependendo do tamanho da estrada. Houve uma necessidade da obtenção destas *bounding boxes* a partir do OSM, utilizando para isso o serviço Nominatim, pois as estradas possuem coordenadas que não são iguais às coordenadas que são fornecidas pelos serviços de navegação. Deste modo, é verificado pela aplicação se as coordenadas obtidas pelos serviços de navegação se encontram dentro destas *bounding boxes* possibilitando assim saber se alguma via de difícil circulação (devido à largura da via) se encontra no trajeto/rota traçado.

4.5.3.3 Tabela polygonpoints

A tabela polygonpoints (Figura 20) possui as coordenadas geográficas, ou nós, referentes a cada via da cidade do Funchal. Embora não esteja sendo utilizada, serviu como referência para testes, nomeadamente para a verificação das coordenadas retornadas pela Directions API da MapQuest. Foi concluído que, embora utilizando a mesma fonte de dados (OSM), que o serviço Nominatim e a Directions API da MapQuest retornavam diferentes coordenadas para as mesmas vias, inviabilizando assim a utilização destes dados para a verificação dos trajetos que tivessem vias estreitas. Outra das

⁴⁷ <http://nominatim.org/>

conclusões obtidas com estes testes foi que, a utilização de *bounding boxes* seria a melhor solução para determinar se as coordenadas retornadas para as rotas traçadas pelos utilizadores no mapa da aplicação se encontravam numa determinada via.

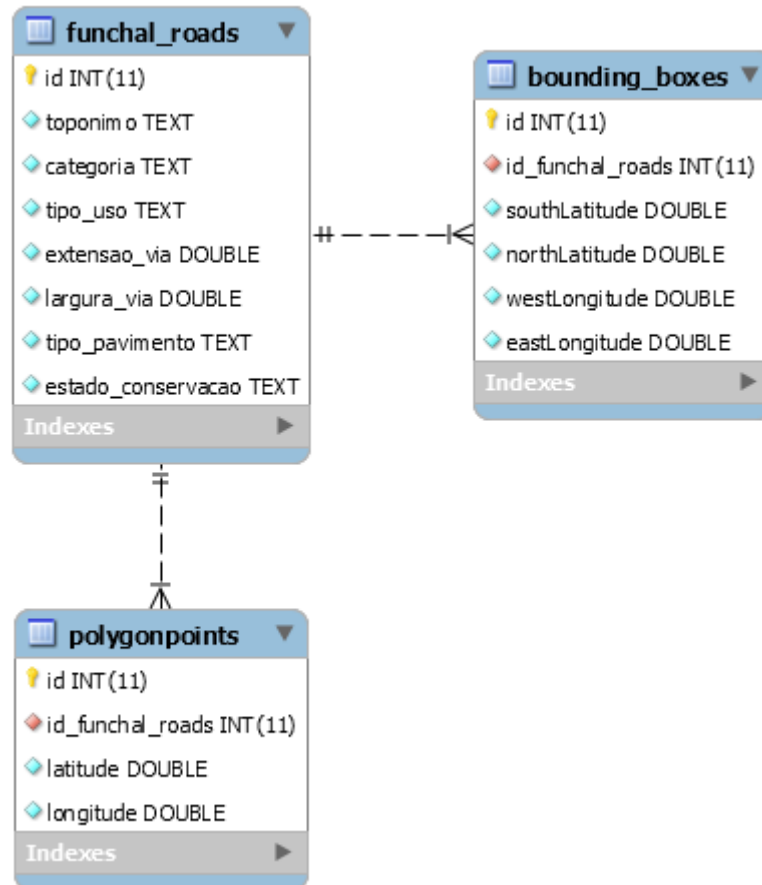


Figura 20: Diagrama Entidade - Relação entre as três tabelas onde são guardadas as informações relacionadas com a largura das estradas.

4.6 Plataforma Android

A plataforma Android foi dividida em diversos módulos, de modo a que cada módulo fosse responsável apenas por alguma parte da execução/obtenção de dados. A plataforma teve como base a *framework* Aware⁴⁸, *framework* que permitia o acesso aos sensores e posterior armazenamento desses dados em bases de dados locais. Foi decidido pela utilização desta *framework* devido a esta fornecer diversas funcionalidades, tais como a obtenção dos dados dos sensores presentes nos telemóveis, posterior tratamento desses mesmos dados e por fim pelo armazenamento

⁴⁸ <https://awareframework.com/>

desses dados já tratados. Alguns dos módulos referidos posteriormente correspondem a diversos *plugins* que são fornecidos por esta *framework*. Devido à plataforma ter sido inicialmente desenvolvida tendo esta *framework* como base, e tendo em conta a estrutura modular que esta apresenta, foi decidido que a implementação das restantes funcionalidades seria feita utilizando a mesma estrutura. Quer os módulos, quer os *plugins* nele inseridos, são compostos por diversos serviços (*Services*) que irão permitir com que a aplicação Android possa obter tanto os dados provenientes dos sensores presentes nos *smartphones*, como os dados que posteriormente serão obtidos a partir do servidor da plataforma ou através de serviços externos à plataforma, como é o caso da Directions API da MapQuest (Route – GET) ou do Google Maps Elevation API⁴⁹.

4.6.1 Módulo Activity Recognition

O *plugin* Google Activity Recognition⁵⁰ fornecido pela *framework* Aware é utilizado para enviar notificações personalizadas aos utilizadores, ao informar se os utilizadores estão caminhando, correndo, andando de bicicleta ou de automóvel. Quando a aplicação se encontra em utilização, as notificações passam a ser recebidas em modo áudio. Por exemplo, se o utilizador se encontra a conduzir, a aplicação envia notificações áudio em vez de notificações de texto de modo a não perturbar a sua condução.

Este *plugin* processa os sinais do dispositivo de modo a identificar as atividades que o utilizador se encontra a realizar em determinado momento. Ao identificar as atividades do utilizador, este *plugin* ajuda a aplicação a melhor se adaptar às necessidades dos utilizadores.

O *plugin* deteta com precisão a atividade dos utilizadores utilizando sinais de baixa potência, lidos através de vários sensores presentes no *smartphone*. As atividades são detetadas automaticamente ao ler os dados dos sensores periodicamente e processando esses mesmos dados utilizando modelos de *machine learning*. Por exemplo, se um utilizador começa a correr, a aplicação recebe uma *callback* indicando a atividade mais provável (correndo).

Reduzir o consumo de memória é importante de modo a manter os *smartphones* a funcionar sem problemas. Ao utilizar o Google Activity Recognition, a aplicação monitoriza as atividades em segundo plano e

⁴⁹ <https://developers.google.com/maps/documentation/elevation/start>

⁵⁰ <https://developers.google.com/location-context/activity-recognition/>

executa uma ação quando uma atividade específica (*in vehicle*) é detetada. A aplicação especifica uma *Pending Intent Callback*, um *Intent Service* que será chamado quando as atividades forem detetadas. Portanto, não existe a necessidade de manter o serviço sempre em execução em segundo plano.

Minimizar o consumo de bateria também é importante. Neste caso a bateria é conservada ao interromper a leitura das atividades quando o dispositivo se encontra parado durante um período prolongado. Este irá retomar as leituras assim que o dispositivo se mova de novo.

4.6.2 Módulo Closed Roads

O módulo `closed_roads` é utilizado para guardar e apresentar todas as informações obtidas a partir do Serviço Regional de Proteção Civil da Madeira. Este módulo utiliza a *framework* `Aware` para armazenar e gerir todas estas informações assim como facilitar a sua obtenção a partir da API fornecida pelo servidor `Node.js`, utilizando para isso a biblioteca `Socket.io`.

A biblioteca `Socket.io` permite também o envio dos dados relativos ao dispositivo que se está conectando ao servidor, o que permitirá posteriormente fornecer a informação relacionada com o último acesso ao servidor. Estes dados serão utilizados assim, na filtragem dos dispositivos que não estivessem ligados à Internet no momento de algum *broadcast* realizado pelo servidor, não possuindo portanto as últimas informações relacionadas com as vias encerradas presentes no servidor, necessitando deste modo de proceder à sua atualização, recebendo no momento de acesso ao servidor os dados que se encontram em falta.

Deste modo este módulo é o responsável por mostrar no mapa todos os troços de estrada encerrados ou condicionados, assim como pelo envio de notificações acerca das estradas que se encontrem em obras ou de eventos que irão ocorrer no futuro.

4.6.3 Módulo Elevations

O módulo `elevations` baseia-se no Google Maps Elevation API que permite obter a elevação de cada uma das coordenadas pertencentes à rota traçada pelo utilizador. A elevação é então obtida a partir de cada uma das coordenadas, realizando o seguinte cálculo a cada 100 metros de distância (aprox.). É calculada a diferença entre a elevação das duas coordenadas e ao valor dessa diferença é então dividida a distância (aprox. 100 metros).

Posteriormente, o resultado desta divisão é multiplicado por 100% de modo a obter a inclinação de um troço de estrada (Figura 21). Deste modo, cada vez que o utilizador escolha uma rota, será mostrado no mapa os troços das vias que possuam uma maior inclinação.

Este processamento é realizado diretamente em cada aplicação móvel e não no servidor, pois existe a necessidade de conhecer as coordenadas e a distância entre as mesmas, o que até poderia ser realizado no servidor, mas que acrescentaria um possível ponto de falha, não apresentando assim nenhuma vantagem. Outro dos problemas, é que para o mesmo trajeto diferentes coordenadas podem ser retornadas, o que impossibilita que estas sejam guardadas na base de dados e que posteriormente sejam utilizadas sem existir a necessidade de recorrer à API.

Existiria também a possibilidade de utilizar os sensores do *smartphone*, para obter a inclinação das estradas, mas que embora apresentassem dados fiáveis e detalhados, necessitariam de ser calibrados de forma a que estes dados pudessem realmente fornecer informação fidedigna. Outro dos problemas seria que os condutores apenas saberiam que uma determinada estrada é inclinada depois de começarem a circular nela, o que invalidaria a grande vantagem da implementação que foi realizada, uma vez que o mecanismo desenvolvido permite saber que estradas são inclinadas com antecedência, sendo diretamente apresentadas no mapa no momento em que o utilizador traça o trajeto para o seu destino no mapa da aplicação.

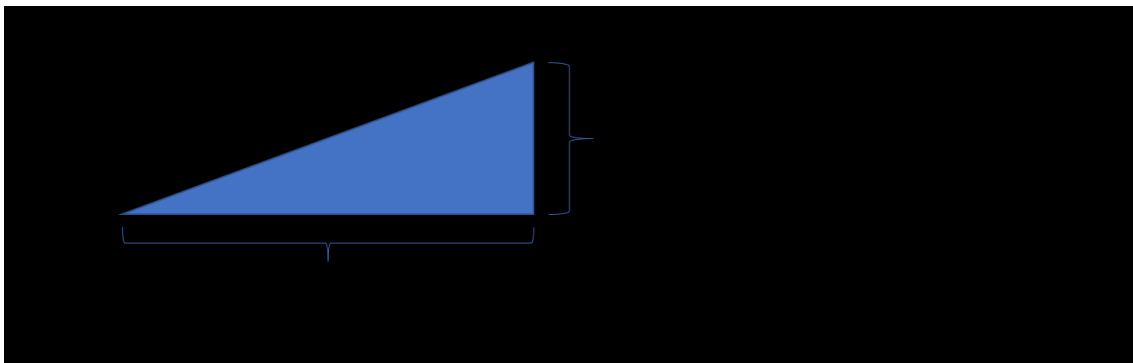


Figura 21: Exemplo da obtenção da inclinação nos troços das vias.

Assim, a fórmula utilizada para a obtenção da inclinação é:

Este cálculo pode por vezes apresentar falso-positivos devido à curvatura das vias e também devido às coordenadas devolvidas não se encontrarem sempre às mesmas distâncias. Por vezes podemos ter vias em que existem coordenadas muito próximas umas das outras, mas em outros

casos podemos ter coordenadas que se distanciem dezenas ou centenas de metros. De qualquer modo, é tida em conta a coordenada correspondente ao ponto inicial onde o utilizador se encontra, e posteriormente a primeira coordenada que se distancie a pelo menos 100 metros de distância da coordenada inicial. Depois disso os cálculos são sempre feitos em relação à última coordenada utilizada no cálculo anterior.

O valor mínimo utilizado para apresentar na aplicação foi uma inclinação de 25%, o que pode ser traduzido numa inclinação de 25 metros por cada 100 metros percorridos, ou seja um rácio de 25:100. Esta percentagem corresponde, no fundo, a uma inclinação de 14.04 graus. Foi utilizado como medida mínima de distância o valor de 100 metros de modo a diminuir o erro causado pelas curvaturas nas estradas, construídas de modo a diminuir a inclinação e principalmente de modo a melhorar o tráfego automóvel. As curvas, tão conhecidas pelos madeirenses, encontram-se em quase todas as estradas da ilha e servem para facilitar a condução, visando deste modo reduzir os níveis de inclinação existentes e facilitar o acesso a algumas zonas da ilha.

Por fim, este módulo é o responsável por informar os utilizadores acerca das vias que possuem uma grande inclinação, dando assim a possibilidade ao condutor de optar por vias alternativas para chegar ao seu destino. Estes troços de estrada inclinados são mostrados diretamente no mapa da aplicação sobre os trajetos traçados pelos utilizadores. Como referido anteriormente, estes troços são calculados e mostrados ao utilizador quando este define a rota para o seu destino, permitindo com antecedência decidir se deseja transitar por uma determinada via ou não.

4.6.4 Módulo WiFi

O módulo WiFi permite ativar a localização por Wi-Fi e permite desativar a localização por satélite sempre que esteja disponível e seja possível a conexão via Wi-Fi. Deste modo evita-se um maior consumo de energia, uma vez que a localização por satélite gasta muito mais energia do que a localização por Wi-Fi.

Este módulo permite também associar as redes Wi-Fi a localizações específicas, neste caso a casa do utilizador e o seu local de trabalho/estabelecimento de ensino. Para isso são guardados todos os acessos a redes Wi-Fi numa base de dados no *smartphone* e contabilizados todos os acessos a essas mesmas redes Wi-Fi. A primeira rede Wi-Fi a chegar a 30 acessos diários (apenas é contabilizado um acesso por dia, mesmo que o

utilizador desconecte da rede Wi-Fi e volte a se conectar posteriormente) será identificada como sendo a casa do utilizador e a segunda rede Wi-Fi a contabilizar 30 acessos diários será considerada como o local de trabalho/estabelecimento de ensino do utilizador.

Após diversos testes, este mecanismo conseguiu identificar com sucesso a casa e o local de trabalho dos utilizadores. No entanto, há que ressaltar diversas situações que poderão levar a que as localizações que venham a ser definidas como casa ou local de trabalho, não sejam informações fidedignas como serão os casos em que o utilizador não possua Wi-Fi em casa ou que o mesmo não tenha ou não utilize esta rede no seu local de trabalho.

4.6.5 Módulo Roads Width

O módulo `roads_width` permite obter as informações relacionadas com a largura das estradas a partir do servidor. Sempre que uma nova rota é traçada são apresentados no mapa todos os troços de estrada que possuam uma largura inferior a 3 metros. Deste modo é possível mostrar aos utilizadores todas as estradas estreitas existentes durante o seu trajeto, permitindo assim que estes consigam evitar estas estradas ou que ao menos se encontrem conscientes de que estas existirão durante a rota para o seu destino.

4.6.6 Módulo Ocorrências

O módulo ocorrências permite obter todas as informações relacionadas com acidentes (*tag* acidente), postadas no grupo Ocorrências Madeira no Facebook, utilizando para isso a API fornecida pelo Facebook. Estas informações foram fornecidas pela empresa Dobsware⁵² que autorizou a utilização desta API para o caso em estudo. Neste módulo é apresentado o tipo de evento, a respetiva data e hora e a informação textual descrevendo o evento que foi postado pelos utilizadores. No caso de ser adicionada alguma imagem ao *post* esta é mostrada ao utilizador, caso não exista nenhuma então é mostrada uma imagem por defeito. As informações obtidas são mostradas então de duas formas: de forma textual e caso existam fotos estas são exibidas na vista Ocorrências Madeira e através da sua geolocalização, com a identificação destas mesmas ocorrências no mapa da aplicação.

⁵² <http://dobsware.com/>

Devido a esta API ter sido desenvolvida por terceiros, e também por esta depender da partilha de informações por parte de utilizadores, esta acaba por ter algumas limitações, nomeadamente na quantidade e qualidade dos dados obtidos. Uma das predefinições existentes, aplicada pelos desenvolvedores da Dobsware, foi de que sempre existiriam no mínimo 6 ocorrências, não tendo em conta quando estas haviam sido postadas. Esta predefinição foi imposta de modo a sempre terem alguma informação presente nas suas próprias aplicações, para que o utilizador não ficasse com a ideia de que os dados não estavam a ser recolhidos. Esta elevada dependência em relação às postagens dos utilizadores, levará a que existam dias com um grande número de ocorrências, enquanto que noutros poderá nem existir nenhuma ocorrência partilhada pelos utilizadores. Como foi descrito anteriormente, será então possível a visualização na aplicação, de ocorrências que tenham acontecido a alguns dias atrás.

4.6.7 Módulos Routes e Instructions

Os módulos routes e instructions são os dois módulos que permitem a navegação *turn-by-turn* utilizando para isso o serviço Directions API da MapQuest (Route – GET). Esta API é gratuita tendo um limite máximo de 15000 *requests* por mês, e necessita de subscrição caso este limite seja ultrapassado. Quando o utilizador cria uma rota de navegação, o módulo routes fica responsável pela obtenção e *parsing* dos dados a partir do pedido ao serviço. Posteriormente, todas as instruções, tais como manobras e respetivo texto informativo são passados ao módulo instructions. Durante a navegação o módulo routes ficará responsável pela navegação em si, mostrando a posição do utilizador no mapa e a respetiva rota a seguir, enquanto que o módulo instructions ficará responsável por mostrar as manobras a serem realizadas (representado pelas setas), o texto informativo com a indicação do nome das estradas e respetivas manobras e também por gerar o áudio dessas mesmas manobras alguns metros antes de o condutor chegar, por exemplo, a algum cruzamento numa via (Figura 22).

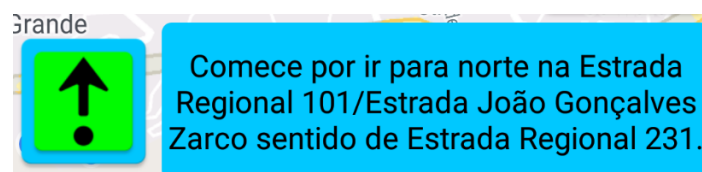


Figura 22: Exemplo da apresentação das manobras a realizar.

4.6.8 Interface de Utilizador

Neste capítulo serão apresentadas as diferentes informações que serão disponibilizadas ao utilizador através da aplicação Android. A maioria delas é apresentada diretamente no mapa, quer seja de forma espontânea sempre que o utilizador abra a aplicação (estradas encerradas/condicionadas e ocorrência de acidentes), pela escolha de um trajeto (estradas com elevado grau de inclinação e estradas estreitas) ou então através da vista onde são apresentados os eventos que são recolhidos tendo como fonte os *posts* introduzidos no Ocorrências Madeira.

Embora vários dos serviços utilizados tenham como base o OSM, e existindo a possibilidade de utilizar o mapa fornecido pelo OSM na aplicação, foi optado pela utilização do mapa da Google, uma vez que inicialmente chegaram a ser testados alguns serviços da Google. Devido ao menor número de *requests* mensais permitidos (comparado com a MapQuest) e devido aos serviços fornecidos pelas diversas entidades relacionadas com o OSM, foram descartados muitos dos serviços da Google. No entanto, não chegou a ser colocada em causa a mudança do mapa da aplicação, uma vez que em nada influenciava o modo como eram apresentados os dados.



Figura 23: Exemplo de estrada encerrada.

Na Figura 23 é apresentada uma estrada encerrada no mapa. O símbolo escolhido foi o sinal de sentido proibido e a cor do trajeto é o vermelho. O símbolo foi escolhido tendo em conta outras soluções, que também apresentam a mesma simbologia para estradas fechadas/encerradas. O vermelho foi escolhido tendo em conta a conotação desta mesma cor e

também pela sua sobreposição sobre o mapa sendo assim muito mais fácil de identificar.

Na Figura 24 é apresentada uma estrada condicionada no mapa. O símbolo escolhido foi o sinal de “em construção” e a cor do trajeto é o laranja. O símbolo foi escolhido tendo em conta que muitas das estradas estão condicionadas devido a obras nas vias. A cor laranja foi escolhida tendo em conta a conotação desta com o perigo e também de forma a diferenciar as estradas condicionadas das estradas encerradas. Esta diferenciação é importante pois, ao contrário das estradas encerradas, nas estradas condicionadas o que acontece é que apenas um dos sentidos da via se encontra encerrado ou então estas apenas se encontram encerradas nalgumas horas do dia, tal como acontece quando as entidades competentes fecham algum troço da via devido a limpezas das escarpas ou então devido à reposição da via com um novo tapete de alcatrão.



Figura 24: Exemplo de uma estrada condicionada.

Na Figura 25 é apresentada uma estrada com um elevado grau de inclinação no mapa. O símbolo escolhido foi o sinal de perigo de inclinação elevada e a cor do trajeto é o roxo. O símbolo foi escolhido tendo em conta a sinalização utilizada em Portugal para as estradas com um elevado grau de inclinação. A cor roxa e o tracejado foram escolhidos de forma a serem visíveis sobre o trajeto pretendido, que também se encontra representado no mapa. Apenas os troços do trajeto em que a inclinação é acentuada são representados no mapa. Tendo em conta que a nossa ilha tende a ter vários troços em que existe uma certa inclinação será possível a visualização, durante o trajeto, de diversas estradas inclinadas.

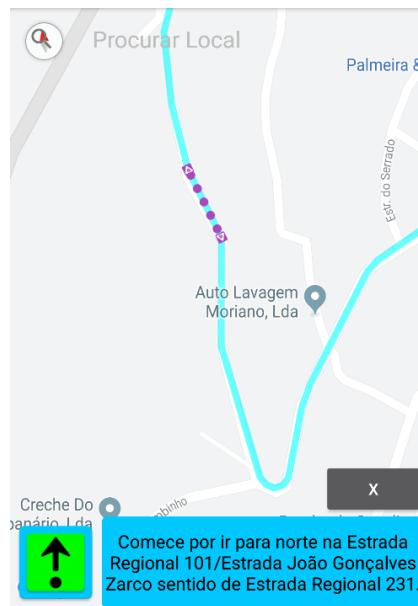


Figura 25: Exemplo de uma via com um elevado grau de inclinação.

Na Figura 26 é apresentada uma estrada estreita no mapa. O símbolo escolhido foi o sinal de perigo de estreitamento da via e a cor do trajeto é o verde. O símbolo foi escolhido tendo em conta a sinalização utilizada em Portugal para as o estreitamento de algum troço de estrada. A cor verde e o tracejado foram escolhidos de forma a serem visíveis sobre o trajeto pretendido, que também se encontra representado no mapa. Apenas os troços do trajeto em que a estrada é estreita (menor que 3 metros) são representados no mapa. As vias estreitas apenas são representadas no concelho do Funchal, pois foi a sua divisão de trânsito que nos forneceu os dados relacionados com a largura das vias.

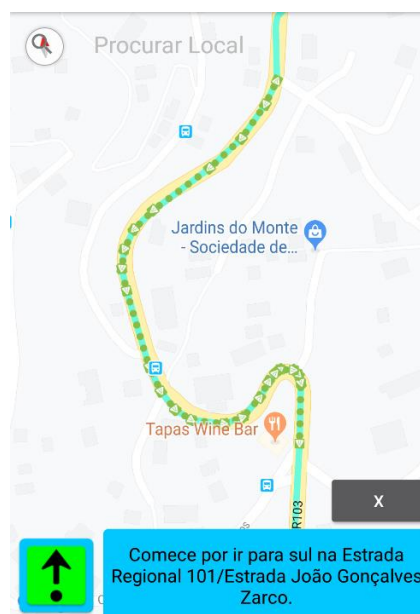


Figura 26: Exemplo de uma estrada estreita.

Na Figura 27 é apresentada a vista em que são exibidos os *posts* do Ocorrências Madeira. Nesta vista é mostrado o tipo de evento ocorrido com a respetiva data, hora e descrição. Esta descrição corresponde à informação textual descrevendo o evento ocorrido e postado pelos utilizadores. No caso de ser adicionada alguma imagem ao *post* esta é mostrada ao utilizador, caso não exista nenhuma então é mostrada uma imagem por defeito. Como é possível constatar, por vezes os dados não são tão atuais, tendo mesmo neste exemplo dados de 3 dias diferentes. Outro dos problemas que podemos detetar facilmente é a irrelevância ou mesmo a inexistência de informação que seja importante para o utilizador. Por exemplo, na primeira ocorrência existe o relato acerca de um acidente no dia anterior, o que para o utilizador não será de extrema relevância. Já na terceira ocorrência não existe nenhuma informação acerca da via nem do local, sendo assim irrelevante a informação fornecida.

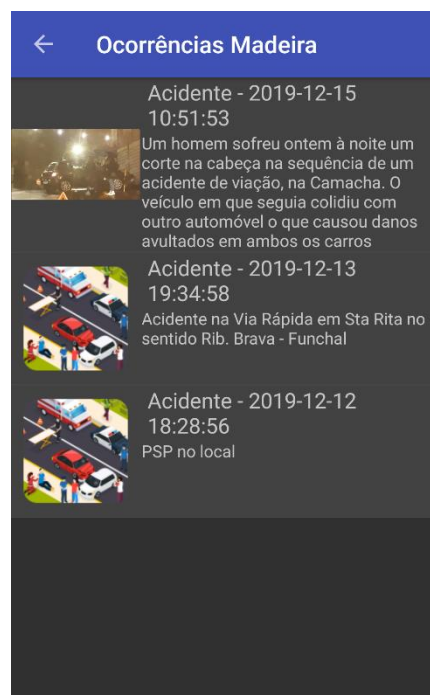


Figura 27: Vista com os dados dos posts do "Ocorrências Madeira".

Na Figura 28 é complementada a informação acima descrita na vista Ocorrências Madeira, ao adicionar no mapa da aplicação a localização da ocorrência, partilhada pelos utilizadores. Embora algumas das informações acima descritas não sejam muito relevantes, é importante conhecer a localização desses mesmos eventos de forma a informar os utilizadores acerca de acidentes que tenham ocorrido no seu trajeto. Deste modo, é apresentado no mapa a localização do acidente. É importante ressaltar que estas informações são partilhadas pelos utilizadores, não havendo a possibilidade de comprovar a veracidade do texto dos *posts* partilhados nem

mesmo da localização fornecida pelos utilizadores. Uma das situações que podem-se verificar é a inexatidão da localização dos acidentes devido aos utilizadores apenas postarem os dados do acidente após pararem as viaturas, o que pode ocorrer em 5 minutos ou numa hora, quer seja perto ou longe do local do acidente.



Figura 28: Exemplo da representação de um acidente partilhado no "Ocorrências Madeira".

Na Figura 29 são apresentados a casa do utilizador e o seu local de trabalho/estabelecimento de ensino, dados que são obtidos a partir do módulo WiFi que foi apresentado acima. Para que seja possível esta apresentação, foram guardados todos os acessos a redes Wi-Fi numa base de dados no *smartphone* e contabilizados todos os acessos a essas mesmas redes Wi-Fi. A primeira rede Wi-Fi a chegar a 30 acessos diários (apenas é contabilizado um acesso por dia, mesmo que o utilizador desconecte da rede Wi-Fi e volte a se conectar posteriormente) será identificada como sendo a casa do utilizador e a segunda rede Wi-Fi a contabilizar 30 acessos diários será considerada como sendo o local de trabalho/estabelecimento de ensino do utilizador. Posteriormente, a rede com mais acessos será considerada sempre como sendo a casa do utilizador, e a segunda como local de trabalho/estabelecimento de ensino. Deste modo, caso o utilizador mude de morada ou de trabalho, a aplicação conseguirá, após algum tempo identificar as novas moradas, quer seja de habitação ou de trabalho.

Atualmente, esta ferramenta também se encontra disponível no Google Maps. Mas, quando o projeto foi iniciado a aplicação do Google não possuía uma forma de identificar automaticamente estas duas localizações. À altura

do início do projeto era possível apenas que o utilizador introduzisse manualmente no mapa a posição onde se encontrava a sua casa e também onde este trabalhava, dados que eram então armazenados pela aplicação.



Figura 29: Exemplo da apresentação dos locais mais frequentados pelos utilizadores (casa e trabalho).

Na Figura 30, após o utilizador criar a rota pretendida, são também mostrados os troços de estrada inclinados ou em que as ruas sejam demasiado estreitas.

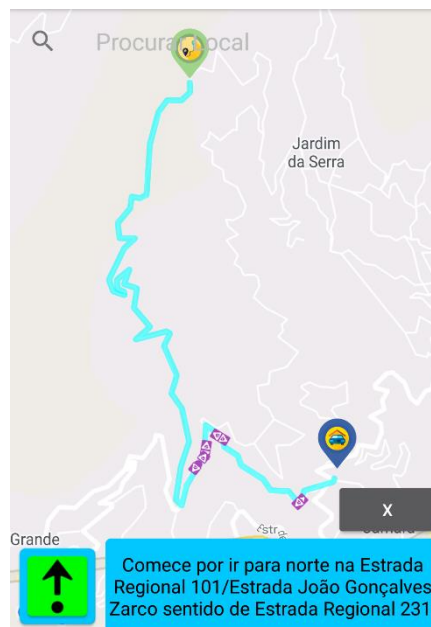


Figura 30: Exemplo da apresentação das manobras a seguir pelo utilizador no mapa.

Para além da rota e respetivos troços inclinados e estreitos, são também apresentados os ícones de início e de destino, que assim iniciam e finalizam

a rota a seguir. Na parte inferior, sobre o mapa, são mostradas as manobras a realizar, numa representação por setas, e ao lado a indicação textual que o utilizador deve seguir, onde são também apresentados os nomes das ruas em que o condutor se encontra ou as ruas em que este deverá virar. Ao mesmo tempo que estas informações são apresentadas, o *text-to-speech* informa o utilizador, de forma sonora, as instruções a seguir.

Acima das instruções existe um botão que permite que os utilizadores eliminem a rota traçada e, caso desejem, criem uma nova rota. De salientar que alguns problemas foram verificados em relação à navegação, nomeadamente em relação à recriação da rota quando o utilizador sai do trajeto desenhado inicialmente.

4.7 Conclusões

Concluindo, a implementação realizada apresenta alguns problemas inerentes à obtenção de dados de terceiros, quer seja devido à necessidade de normalização dos dados, ou até mesmo pela limitação do número de *requests* que poderão ser realizados mensalmente a algumas APIs. No entanto, em relação a este último problema, existiria a possibilidade de escalar a aplicação ao contratar serviços pagos fornecidos pelas empresas proprietárias destas APIs. Por outro lado, devido à contenção de custos sempre foram optadas pelas soluções gratuitas, e também por aquelas que embora pagas, fornecessem serviços gratuitos com um limite de *requests* mensais em que fosse possível testar com alguns utilizadores. Por esta razão, foram utilizados muitos dos serviços da MapQuest, que oferece 15000 *requests* mensais, e que para serviços semelhantes a Google oferece 5 vezes menos (3000 *requests*).

Outro dos problemas referidos, que não foi implementado, foi a recriação das rotas caso o utilizador saia do trajeto traçado. Uma solução para este problema exigiria muitos mais testes, e a criação de um algoritmo que recriasse as rotas sempre que o utilizador seguisse numa via que não estivesse na rota traçada.

Outra das soluções que deveriam ser implementadas, caso a aplicação fosse para ambiente de produção seria a compra de um domínio, de modo a não existir a necessidade de utilização do Nip.io. Não porque a solução não resolva os problemas atuais, mas porque a aquisição de um domínio demonstraria um maior profissionalismo, evitando que existisse a necessidade da utilização do IP no domínio.

Por fim, a perda da ligação à Internet, tanto no servidor como na aplicação Android, levaria a que não fosse possível a obtenção e envio de novos dados do servidor para a aplicação Android. Caso esta última não estivesse conectada à Internet, não receberia novos dados, mas todos os que já se encontrassem nas bases de dados locais, seriam mostrados aos utilizadores. Assim, a perda de conexão à Internet não invalidaria a utilização da aplicação, uma vez que esta possui muitos dados guardados, mas limitaria, e muito, no pedido de novos dados ou na criação de novas rotas.

A solução apresentada tem uma boa robustez, apenas pecando nos pontos acima mencionados. De salientar, que ao serem utilizados serviços gratuitos, a probabilidade de encontrarmos servidores em *downtime* ou sobrecarregados é muito superior do que se estivéssemos utilizando uma solução paga que nos permitiria aceder a servidores com uma grande percentagem de tempo em *uptime*.

5. Testes e Resultados

Este capítulo vem introduzir alguns dos testes efetuados durante o desenvolvimento da aplicação, sendo que estes são importantes de modo a descobrir possíveis erros ou problemas com a GUI que não foram detetados durante o desenvolvimento da plataforma, e também possíveis problemas que possam vir a aparecer nas APIs utilizadas pelo servidor para a obtenção dos dados.

Primeiramente serão apresentados os tempos de acesso às APIs no servidor, testes que foram armazenados em ficheiros de *log*, possuindo os *timings* necessários para um conjunto de *requests*. Toda a vez que um *request* foi feito foi guardado o tempo de resposta nestes ficheiros de *log*, de modo a ser possível identificar os serviços que pudessem vir a apresentar problemas.

Posteriormente serão apresentados os testes dos utilizadores, onde alguns utilizadores instalaram a aplicação e usaram-na durante algum tempo, de modo a ser possível identificar possíveis problemas e respetivas melhorias a serem feitas. De salientar, que foi deveras importante a realização destes testes, pois permitiram tomar conhecimento de alguns problemas na plataforma e também permitiram obter outras perspetivas e ideias acerca de como a aplicação poderia ser melhorada no futuro.

5.1 Tempos de Acesso (APIs) no Servidor

Foram realizados testes aos tempos de acesso do servidor às APIs que são utilizadas para a obtenção de dados de geolocalização, mas também em relação à obtenção dos dados relativos ao *website* do Serviço Regional de Proteção Civil da Madeira. Todos estes testes foram baseados em *logs* guardados no servidor contendo os tempos de resposta aos *requests* efetuados. Todos os tempos de resposta foram guardados, e apenas uma amostra desses dados foram tidos em conta (cerca de 100 *requests*), de forma a apresentar a existência de problemas (p. ex. servidor em *downtime* ou sobrecarregado) ou não, relacionados com esses pedidos.

5.1.1 Proteção Civil

Os dados obtidos a partir do *website* do Serviço Regional de Proteção Civil da Madeira, são obtidos a partir do ficheiro KML, onde é realizado um *parsing* e retirados todos os dados necessários e relevantes para a plataforma.

De salientar que estes dados são relativos à implementação do último algoritmo para a obtenção dos dados a partir deste *website*, uma vez que anteriormente era realizado um *parsing* HTML, que retirava os dados diretamente a partir do DOM. Este *parsing* era muito mais demorado e também era muito mais vulnerável em relação a falhas pois exigia que não existissem mudanças no esquema em que o DOM estava organizado.

O gráfico com os tempos de duração de cada pedido é apresentado na Figura 31, com dados retirados desde o dia 1 de maio de 2019 até ao dia 7 de agosto de 2019. Nesta amostra foram realizados 100 pedidos, com uma média de 87,19 milissegundos, tendo como mínimo um tempo de 69 milissegundos e como tempo máximo 236 milissegundos. Estes tempos podem ser considerados como muito bons, o que garante com que seja afirmado, com uma certa garantia, que o pedido de dados do *website* do Serviço Regional de Proteção Civil da Madeira possam ser obtidos com uma taxa de 100% de sucesso.

Estes tempos de resposta são importantes, pois permitiram identificar que o *website* esteve sempre disponível e que foi possível realizar a obtenção dos dados todos os dias. Por outro lado, se o *website* não estivesse disponível seria impossível a obtenção dos dados e posterior apresentação no mapa da aplicação Android.

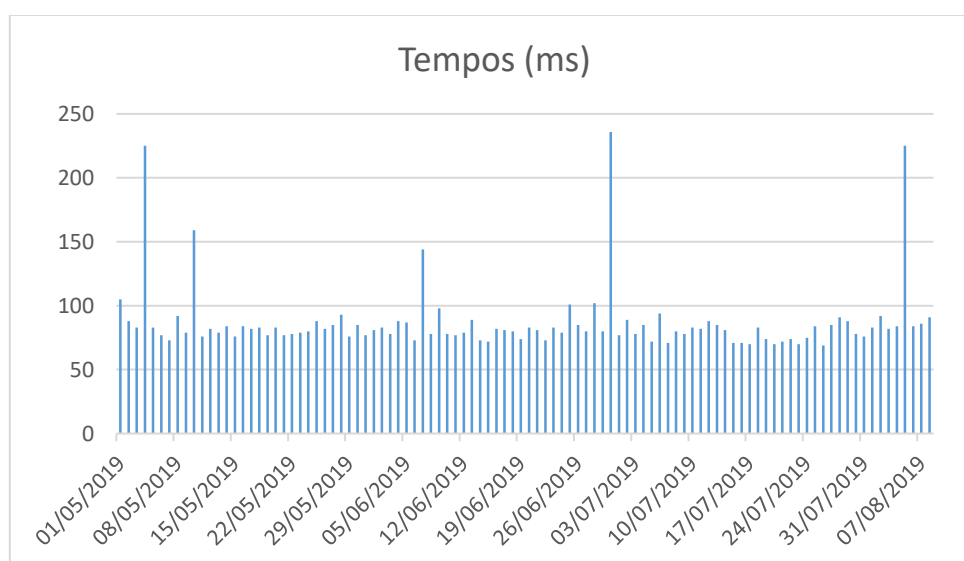


Figura 31: Gráfico com os tempos de acesso ao *website* do Serviço Regional de Proteção Civil da Madeira.

5.1.2 Overpass API (Obtenção do nome da via encerrada)

O gráfico com os tempos de duração de cada pedido é apresentado na Figura 32, com dados retirados no mês de agosto. Nesta amostra foram realizados 100 pedidos, com uma média de 417,41 milissegundos, tendo

como mínimo um tempo de 151 milissegundos e como tempo máximo 2553 milissegundos. Estes tempos podem ser considerados como razoáveis, o que garante com que seja afirmado, com uma certa garantia, que a obtenção dos dados relacionados com os pedidos do nome das vias encerradas possam ser obtidos com uma taxa de 100% de sucesso.

No entanto, devido a existirem alguns pedidos que acabaram por demorar mais algum tempo, poderão existir alguns deles que venham a falhar. Isto, como já foi descrito acima, acontece devido a este serviço ser completamente gratuito, o que faz com que muitas vezes os servidores estejam sobrelotados e tenham de atrasar as respostas aos pedidos ou até cancelar os mesmos.

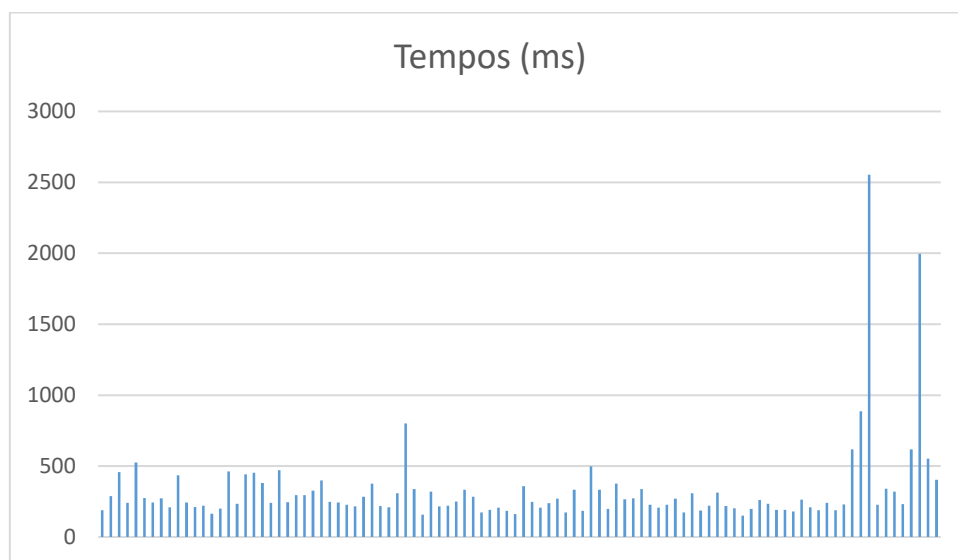


Figura 32: Gráfico com os tempos de acesso à Overpass API (Obtenção do nome da via encerrada).

5.1.3 Overpass API (Obtenção dos nós de cada via)

O gráfico com os tempos de duração de cada pedido é apresentado na Figura 33, com dados retirados no mês de agosto. Nesta amostra foram realizados 100 pedidos, com uma média de 2604,42 milissegundos, tendo como mínimo um tempo de 570 milissegundos e como tempo máximo 11428 milissegundos. Estes tempos podem ser considerados como abaixo do razoável, o que não garante que tenhamos sempre 100% de sucesso nos pedidos realizados. Embora durante os testes, e durante a utilização da plataforma isto não tenha acontecido, devido ao volume de dados que são obtidos e à exigência da *query* em si, não podemos assim garantir o sucesso de todos os pedidos. Tal como o outro pedido ao servidor do Overpass API,

este depende da disponibilidade do servidor, sendo que os problemas derivados deste serviço já foram apresentados anteriormente.

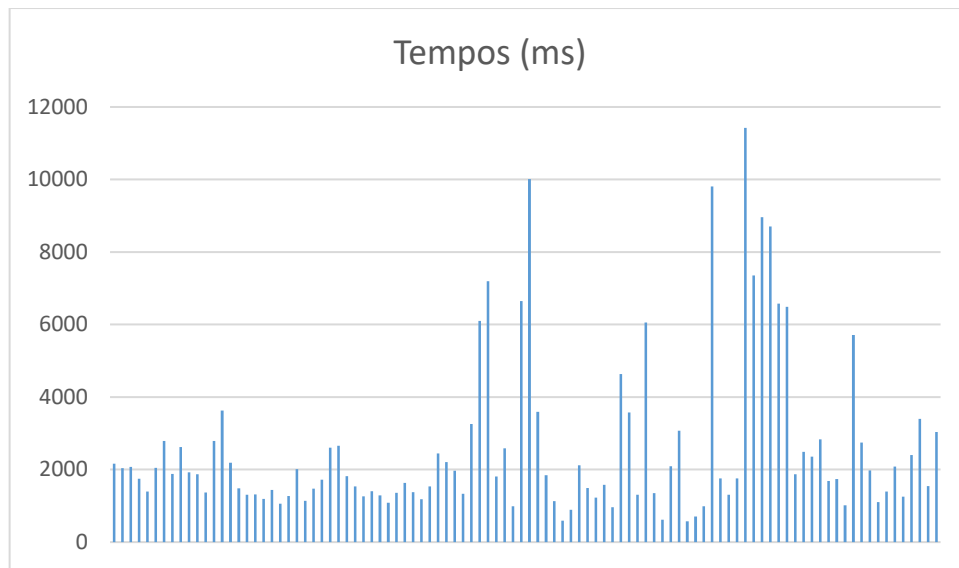


Figura 33: Gráfico com os tempos de acesso à Overpass API (Obtenção dos nós de cada via).

5.1.4 MapQuest API (Obtenção dos Link Ids)

O gráfico com os tempos de duração de cada pedido é apresentado na Figura 34, com dados retirados no mês de agosto. Nesta amostra foram realizados 100 pedidos, com uma média de 476,67 milissegundos, tendo como mínimo um tempo de 227 milissegundos e como tempo máximo 562 milissegundos.

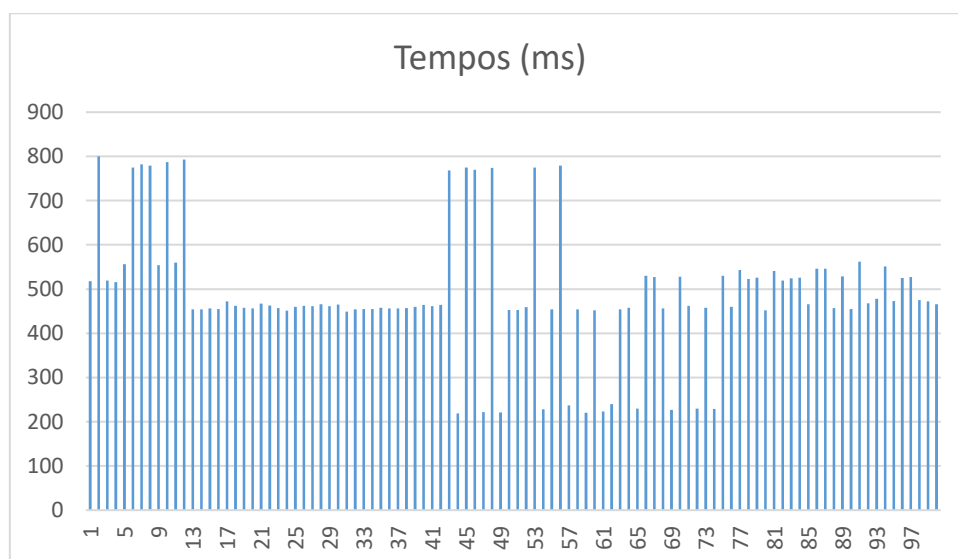


Figura 34: Gráfico com os tempos de acesso ao MapQuest API (Obtenção dos Link Ids).

Estes tempos podem ser considerados como muito bons, o que garante com que seja afirmado, com uma certa garantia, que os Link Ids possam ser obtidos com uma taxa de 100% de sucesso.

Esta API é deveras importante pois permitirá obter os Link Ids para cada início e fim de via encerrada, sendo posteriormente utilizados como valores do parâmetro `mustAvoidLinkIds` nas rotas traçadas na aplicação. Estes valores permitirão assim traçar rotas evitando os troços de estrada que se encontrem encerrados. A MapQuest disponibiliza assim uma forma de podermos traçar estas rotas evitando determinados troços de estrada, sendo que, no entanto, existe a necessidade da obtenção destes Link Ids, uma vez que não existe a possibilidade de passar os valores das coordenadas nos serviços de navegação.

Cada Link Id é correspondente a um troço de via, mas não necessariamente a uma via completa, existindo assim a necessidade da obtenção dos Link Ids de início e de fim das vias encerradas. Por vezes o Link Id poderá ser o mesmo caso seja uma via de pequena dimensão, mas o mais comum é que exista mais do que um Link Id para cada via. Cada vez que necessitemos de um Link Id é necessário realizar um *request*, não existindo a opção de passar um *bundle* de coordenadas, podendo assim obter diversos Link Ids no mesmo pedido.

Concluindo, os serviços da Google e da MapQuest têm tempos de resposta muitos bons, mas os serviços do Overpass API têm tempos de resposta algumas vezes superiores ao desejável, devido à elevada utilização dos servidores do OSM, serviço em que se baseia o Overpass API. No entanto, por ser gratuito e permitir a pesquisa de uma elevada quantidade de dados, optou-se por este serviço.

Outro dos motivos da obtenção dos dados a partir do Overpass API tem a ver com a necessidade da obtenção de dados específicos, nomeadamente os nós com as coordenadas de cada via, que de outra forma seria mais difícil ou impossível de obter, ou então, não seriam gratuitos.

5.2 Atributos e Cenários de Qualidade

Neste subcapítulo serão apresentados os resultados referentes à implementação dos atributos de qualidade propostos na secção 3, que aborda a arquitetura de software da plataforma. Embora a implementação de alguns destes atributos já tenha sido abordada anteriormente noutros capítulos, é

deveras importante explicar detalhadamente o que foi feito e de que modo foi implementado cada atributo de qualidade.

5.2.1. Disponibilidade

Em relação à disponibilidade, foi desenvolvida uma base de dados onde foram armazenados os dados provenientes do servidor, nomeadamente os dados provenientes do Serviço Regional de Proteção Civil da Madeira. Caso a aplicação não possuísse ligação à Internet, seria sempre possível mostrar os dados atualmente existentes na base de dados da aplicação.

Por outro lado, os dados obtidos a partir do Ocorrências Madeira, são obtidos diretamente a partir da aplicação presente no *smartphone*, mas caso não exista acesso à Internet, não será possível a obtenção dos dados. A utilização desta estratégia deveu-se a que os pedidos de dados do Ocorrências Madeira devessem ser obtidos em tempo real, uma vez que se tratava de eventos partilhados pelos utilizadores e que poderiam apenas ter alguma relevância se fossem atualizados de forma consistente.

De outro modo, caso o pedido fosse feito apenas pelo servidor, e depois realizado o *broadcast* destes dados para todos os dispositivos móveis, estaríamos a adicionar um ponto de falha (servidor). Assim, cada aplicação irá receber os dados mais recentes que tenham sido postados no grupo Ocorrências Madeira, sem que tenha de ser feito algum pedido ao servidor.

Outra das razões relacionadas com a não implementação do pedido desta API no servidor, foi por, no fundo, pelo que foi explicado pelos desenvolvedores da Dobsware, existir um limite de 100000 *requests*, o que é mais do que necessário para a quantidade de utilizadores que testaram a aplicação. Caso a aplicação passasse para um ambiente de produção, teria de ser contratado um serviço semelhante ao que atualmente existe, ao Facebook, de modo a podermos acomodar o possível crescimento do número de utilizadores a usar a aplicação.

5.2.2. Segurança

Em relação à segurança, foram tidas em conta diversas variáveis, como a encriptação dos dados fornecidos pelo servidor, utilizando SSL através do protocolo HTTPS, bem como a encriptação dos dados de *login* do administrador de sistema na base de dados do servidor. Todas estas soluções

foram devidamente testadas de modo a que fosse possível dar a maior segurança e privacidade aos utilizadores no momento de usarem a aplicação.

5.2.3. Desempenho

Em relação ao desempenho, foram encontrados alguns problemas ao longo do desenvolvimento do projeto, que foram prontamente resolvidos. No entanto, quando dependemos da qualidade da Internet, podemos ter alguns problemas relacionados com o desempenho. Mas, em condições normais nenhum problema foi encontrado ao utilizar a aplicação e a obter os dados provenientes do servidor.

5.2.4. Modificabilidade

Em relação à modificabilidade, tanto em termos de código, como em relação à base de dados será possível de forma relativamente fácil a modificação de cada um destes componentes. Em relação ao código, todo ele foi construído em módulos focando na separação de responsabilidades, tanto no servidor como na aplicação.

Já em relação à base de dados, também houve este cuidado, no entanto uma mudança nas colunas da base de dados do servidor poderá obrigar à inserção da mesma coluna na base de dados da aplicação e também à implementação de algum código adicional. Tendo em conta que os dados são partilhados num formato JSON entre o servidor e a aplicação, terá de ser adicionado algum código que realize o *parsing* desta informação extra.

5.2.5. Interoperabilidade

Em relação à interoperabilidade entre a aplicação e o servidor, durante a utilização da plataforma, não existiram falhas entre a troca de mensagens, o que também não foi verificado na obtenção de informações a partir do Ocorrências Madeira.

Já em relação à obtenção de dados a partir de fontes externas por parte do servidor verificou-se algumas falhas especialmente nos pedidos aos serviços do Overpass API. Devido a existirem restrições no número de acessos por segundo e na quantidade de utilizadores a acederem ao mesmo tempo aos seus servidores, foram verificadas falhas que foram parcialmente solucionadas com o aumento do tempo entre os pedidos ao servidor do

Overpass API. No entanto, caso os servidores se encontrem sobrelotados, ou seja, com muitos utilizadores acedendo ao mesmo tempo, será impossível garantir o sucesso dos pedidos. Mas, como foi explicado anteriormente, por ser gratuito e por permitir a pesquisa de uma elevada quantidade de dados, optou-se por este serviço.

5.3 Questionários

Foi desenvolvido um questionário destinado a potenciais utilizadores da aplicação (Anexo D), sendo que o preenchimento do mesmo foi requisitado após algumas semanas de utilização da aplicação, de forma a obter informações relevantes em relação à funcionalidade do sistema em geral e também em relação à GUI.

Foram inicialmente explicadas as funcionalidades da aplicação, mas foi dada a liberdade aos utilizadores de explorar a aplicação de modo a encontrarem eventuais falhas, ou sugestões de melhoria relacionadas com a GUI.

O objetivo do questionário foi compreender quais as funcionalidades que os utilizadores experimentaram durante os testes bem como as dificuldades encontradas durante a utilização da aplicação. Assim é possível obter algum *feedback* em relação ao *design* e por fim tentar compreender se o tipo de informação apresentado é adequado ao fim pretendido.

O questionário é composto por várias perguntas, sendo a maioria destas de resposta obrigatória seguindo a escala de Likert, com a possibilidade de escolher uma opção entre as 5 possíveis. As principais razões da escolha da escala de Likert é porque as perguntas criadas usando esta escala são de mais fácil interpretação e consequentemente poderão ser respondidas mais rapidamente, uma vez que este é um método bastante conhecido. Outro motivo é por esta permitir a extração simplificada de dados [24].

5.3.1. Discussão do resultado dos questionários

Ao analisar o resultado dos questionários concluiu-se que, segundo os inquiridos, a generalidade do *design* e respetivas funcionalidades da aplicação se encontravam implementados de forma adequada, uma vez que os ícones/*design* utilizados eram de fácil interpretação e as várias ferramentas fornecidas de fácil utilização.

Apenas algumas ressalvas foram feitas, nomeadamente em relação aos pontos a seguir enunciados:

- Proximidade dos troços de estrada fechados/condicionados que, principalmente quando existem muitas vias fechadas/condicionadas, se tornam de difícil visualização;
- Por vezes, nas estradas condicionadas, o nome das ruas sobrepõe-se ao troço de via encerrada que é apresentado no mapa;
- A cor verde utilizada para a representação das estradas estreitas não contrasta muito bem com a cor do troço representando a navegação, que é azul;
- Na navegação GPS (*turn-by-turn*) foram encontrados alguns problemas principalmente relacionados com dois pontos, sendo que no primeiro, caso o utilizador mude de vista (p. ex. para a vista definições) a rota anteriormente traçada é perdida, já no segundo, caso o utilizador saia da rota traçada não existe nenhum mecanismo que faça com que o trajeto seja traçado novamente tendo em conta a posição atual do utilizador;
- Também na navegação GPS, as indicações textuais algumas vezes não se encontram com uma sintaxe correta dando a ideia de que o texto foi diretamente traduzido do inglês;
- Em relação à vista onde são apresentados os dados retirados a partir do grupo do Facebook Ocorrências Madeira, existe um problema relacionado com a apresentação de eventos que ocorreram já a algum tempo. Isto deve-se principalmente ao problema de existir por parte dos programadores da Dobsware, proprietários da API utilizada, o intuito de apresentar sempre informações mesmo que elas tenham ocorrido a alguns dias. Deste modo são sempre apresentados no mínimo 6 eventos, de forma a que a vista nunca se encontrar vazia;

Algumas das sugestões dadas pelos utilizadores foram a inserção de uma espécie de *wiki* explicando o que cada ícone apresentado no mapa significa, e a inserção de funcionalidades de navegação GPS semelhantes ao do Google Maps, sendo elas: a aproximação do ícone de navegação quando já nos encontramos a cumprir o trajeto e possibilitar também a criação de trajetos pedonais e de bicicleta. Neste último foi dado o exemplo de que, quando existem eventos desportivos ou celebrações, também se torne mais difícil circular de bicicleta por algumas vias.

5.4 Limitações da Aplicação

A aplicação desenvolvida apresenta um conjunto de limitações associadas por um lado à dificuldade de implementação de algumas funcionalidades, e por outro lado, pela razão de os utilizadores normalmente utilizarem outros sistemas de navegação GPS mais avançados, como por exemplo o Google Maps. No fundo, os utilizadores esperam que a aplicação também forneça muitas das funcionalidades destes sistemas, pois já se encontram acostumados a eles.

Uma das primeiras limitações identificadas foi relacionada com a recriação do trajeto caso o utilizador saia da rota previamente definida. Embora seja possível a implementação de um sistema que seja capaz de recriar esta funcionalidade, torna-se especialmente difícil criar um algoritmo que para além de recriar o trajeto também consiga evitar erros inerentes ao próprio GPS, em que o utilizador é identificado como estando num ponto diferente do mapa ainda que este esteja seguindo o trajeto desenhado de início.

Outra limitação presente no sistema desenvolvido está associada à apresentação dos dados provenientes do Ocorrências Madeira. Neste caso, como a plataforma foi desenvolvida por terceiros, e apenas estando utilizando a API, existem alguns problemas para os quais não existe resolução. Um desses problemas prende-se com a apresentação de eventos que já ocorreram a alguns dias, dados que algumas vezes podem não ser do interesse dos utilizadores. Isto deve-se em grande parte à opção tomada pelos desenvolvedores da Dobsware em mostrar pelo menos 6 eventos, de forma a que a sua aplicação mostrasse sempre alguns eventos, mesmo que eles não tivessem ocorrido no mesmo dia em que a informação é apresentada ao utilizador. Outro dos problemas existentes é relacionado com os dados provenientes do Facebook, em que o conteúdo dos *posts* não é informativo e onde apenas a coordenada do evento é de algum interesse. Nestes casos é apresentado da mesma forma o conteúdo textual do *post* embora este possa apenas ter algum texto, que em nada acrescenta o conteúdo da informação.

Outra limitação identificada é referente à impossibilidade de acesso a determinadas informações na inexistência de ligação à Internet. Esta limitação deve-se ao facto de os dados estarem presentes apenas no servidor e não no cliente (*smartphone*) e também pela utilização de serviços externos que apenas podem ser utilizados com ligação à Internet (p. ex. navegação GPS). A presença de alguns dados no servidor e não no terminal do cliente deveram-se, por um lado, à tentativa de economizar memória nos

smartphones, uma vez que a quantidade de dados existente é relativamente grande, e por outro lado de modo a economizar processamento no lado do cliente, uma vez que o processamento feito pelo servidor evita com que este tenha que ser feito em cada dispositivo móvel.

6. Conclusões

O projeto desenvolvido teve como objetivos iniciais a obtenção de dados através de diversas fontes, podendo deste modo centralizar estes mesmos dados numa única plataforma, e a obtenção da inclinação e da largura das estradas.

Para isso foi criado um servidor que efetua a recolha desses mesmos dados e os envia para a aplicação Android, sendo esta última a responsável por mostrar e informar os utilizadores acerca das condições de tráfego.

Uma das dificuldades sentidas ao longo do projeto foi a recolha de dados a partir de terceiros, uma vez que dependendo da fonte o algoritmo de recolha tinha de ser implementado de forma diferente. Além disso, para a mesma fonte, nomeadamente o *website* do Serviço Regional de Proteção Civil da Madeira, houve uma certa dificuldade devido às constantes mudanças no site, em que inicialmente era feita a recolha através de um *parsing* HTML, que teve de ser mudado devido às diferentes versões, e que acabaria finalmente por utilizar um mapa do Google MyMaps (de onde é retirado o KML onde atualmente é realizado o *parsing*), que facilitou na questão do *parsing*, mas que dificultou no ponto em que o código-fonte teve de ser alterado devido a todas estas mudanças.

Por fim, a aplicação desenvolvida consiste como um complemento às aplicações já existentes no mercado, mas que como são aplicações utilizadas globalmente, não têm em conta os problemas relacionados com a ilha da Madeira. Neste ponto, a aplicação desenvolvida preenche determinadas lacunas que foram identificadas noutras aplicações, tendo em vista o mercado regional ou os turistas que se encontram a passar férias na região.

6.1 Objetivos Atingidos

Tendo em conta os objetivos que foram inicialmente definidos, estes foram parcialmente alcançados. Apenas algo a salientar, é a forma como os dados são recolhidos, e que por vezes necessitam de intervenção humana para uma normalização desses mesmos dados, a partir da página de administração. Várias tentativas de automatizar este processo foram desenvolvidas, mas sem sucesso. As sucessivas alterações do esquema

dos dados a partir das fontes e também o preenchimento pela parte de terceiros desses mesmos dados, levou a que fosse necessário criar a página de administração e a intervenção do administrador de sistema em determinados casos.

Isto tornou-se ainda mais importante quando foi detetado que alguns dos registos relacionados com as vias encerradas não possuíam data de encerramento e/ou reabertura. Após encontrar os editais emitidos pelas próprias câmaras, foi identificado que todas elas tinham uma data de encerramento e que a data de reabertura, quando não existente, era identificada com o tempo de trabalho projetado para os trabalhos naquela via.

6.2 Trabalho Futuro

Como trabalho futuro pretende-se:

- A tentativa de automatização da normalização dos dados, uma vez que se torna incomportável a necessidade de existir sempre alguém a normalizar os dados. Isto requer tempo, e a identificação de um padrão, criando também vários algoritmos que consigam lidar com as constantes mudanças;
- Replicar a base de dados relacional para uma base de dados orientada a objetos ou uma base de dados de grafos, pois tendo em conta os dados que estão sendo armazenados, torna-se evidente que estas duas últimas bases de dados seriam as melhores soluções pois estão mais preparadas para realizar pesquisas em árvores. Uma das possibilidades de utilização de uma base de dados orientada a objetos seria o MongoDB enquanto que para uma base de dados de grafos uma das soluções a ter em conta seria o Neo4j;
- Dependendo do número de utilizadores, caso houvesse um crescimento considerável, poder escalar os recursos do servidor de modo a melhor poder acomodar a carga no sistema. Inicialmente, devido aos custos acrescidos foi optado pelo servidor mais barato tendo em conta a contenção de custos, mas que poderá não suprir as necessidades caso o número de utilizadores aumente consideravelmente;
- Aumentar as fontes de dados, pois atualmente apenas são utilizadas duas fontes de dados que foram as mais importantes que foram encontradas, e que conjugavam os dados das entidades governamentais com os dados partilhados pelos próprios

cidadãos nas redes sociais. Neste caso, terá de ser criado um algoritmo que consiga identificar informações semelhantes, evitando assim mostrar mais que uma vez a mesma informação ao utilizador.

Referências

- [1] M. Macedo, «Análise da Evolução da Rede Rodoviária e das Acessibilidades na Ilha da Madeira», Dissertação de Mestrado, Universidade Nova de Lisboa, Lisboa, 2018.
- [2] A. Mendonça, «População Madeirense num Período de Transição: Uma década (1991-2001) - Viagem pelos Concelhos», *Revista Girão*, 2011.
- [3] D. R. de Estatística da Madeira, «Estatísticas Demográficas da Região Autónoma da Madeira 2017», *Funchal DREM*, 2018.
- [4] S. R. dos Assuntos Sociais, «A Saúde com as pessoas - PRS 2011-2016», *Funchal IP-RAM*, 2011.
- [5] Laboratório Regional de Engenharia Civil, Agência Regional da Energia e Ambiente da Região Autónoma da Madeira, e Câmara Municipal do Funchal, «Caso de Estudo sobre Poluição Atmosférica - Funchal», *Funchal Proj. ISIS*, 1996.
- [6] J. França e A. Almeida, «Plano Regional de Água da Madeira - Síntese do Diagnóstico e dos Objetivos», 6º *SILUSBA – Simpósio Hidráulica E Recur. Hídricos Países Língua Of. Port.*, p. 68, 2003.
- [7] J. Jesus, S. Teixeira, D. Teixeira, T. Freitas, e D. Russo, «Vertebrados terrestres autóctones dos Arquipélagos da Madeira e Selvagens», *Funchal Dir. Reg. Ambiente*, 2009.
- [8] A. Almeida, «Modelling Tourism Demand in Madeira since 1946 and Historical Overview based on a Time Series Approach», *J. Spat. Organ. Dyn.*, vol. 4, n. 2, pp. 145–156, 2016.
- [9] C. Khatri, «Real-time Road Traffic Information Detection Through Social Media», Dissertação de Mestrado em Engenharia Informática e Engenharia Civil, Georgia Institute of Technology, Atlanta, Georgia, 2015.
- [10] E. Alomari, R. Mehmood, e I. Katib, «Sentiment Analysis of Arabic Tweets for Road Traffic Congestion and Event Detection», em *Smart Infrastructure and Applications*, R. Mehmood, S. See, I. Katib, e I. Chlamtac, Eds. Cham: Springer International Publishing, 2019, pp. 37–54.
- [11] Y. Gu, Z. Qian, e F. Chen, «From Twitter to detector: Real-time traffic incident detection using social media data», *Transp. Res. Part C Emerg. Technol.*, vol. 67, pp. 321–342, Jun. 2016.
- [12] S. Ribeiro, C. Davis, D. Oliveira, W. Meira, T. Gonçalves, e G. Pappa, «Traffic Observatory: a system to detect and locate traffic events and conditions using Twitter», em *Proceedings of the 5th International Workshop on Location-Based Social Networks - LBSN '12*, Redondo Beach, California, 2012.

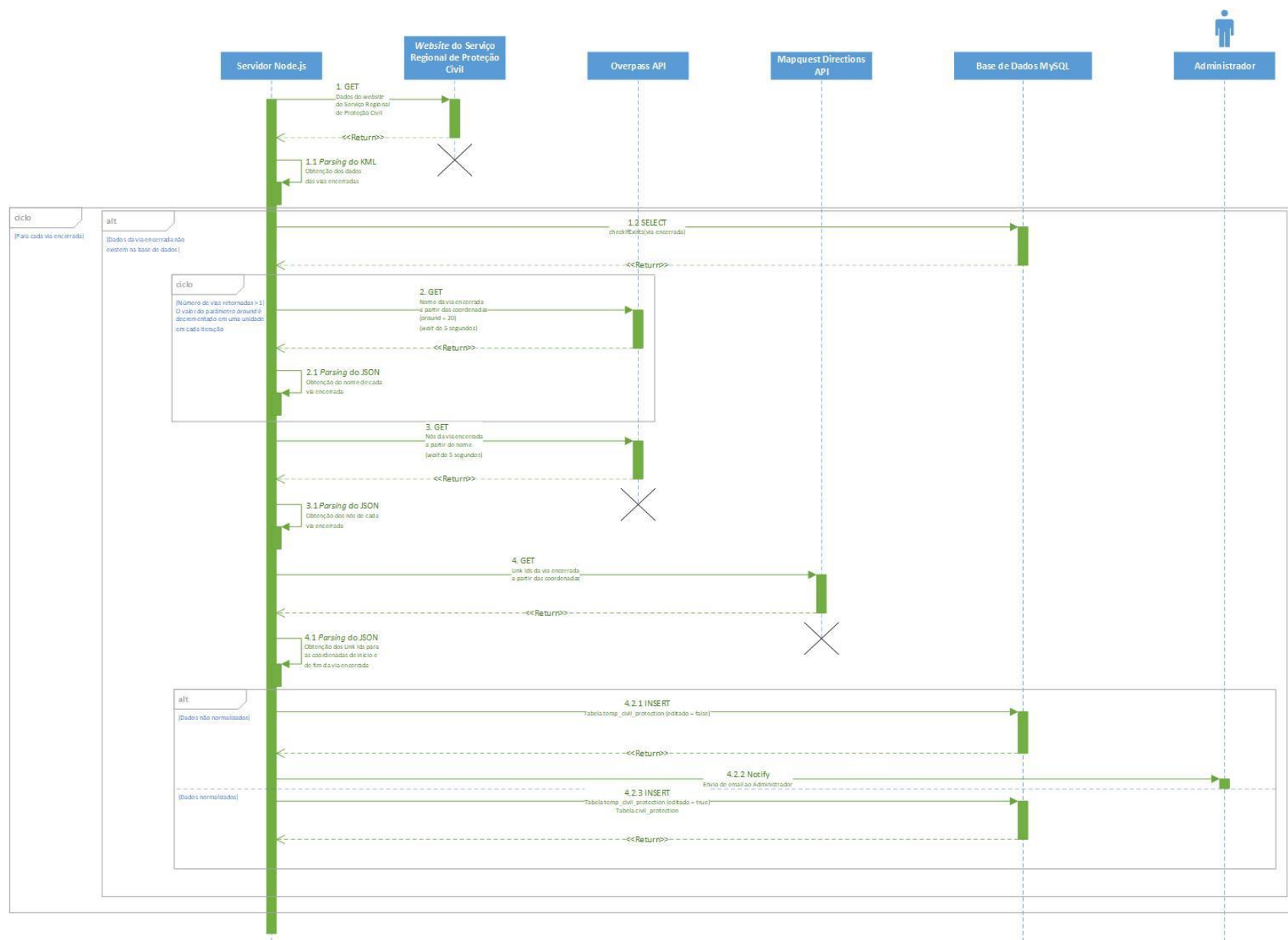
- [13] Z. Zhang, M. Ni, Q. He, e J. Gao, «Mining Transportation Information From Social Media for Planned and Unplanned Events», Mai. 2016.
- [14] S. Chatterjee, S. K. Mridha, S. Bhattacharyya, S. Shakhari, e M. Bhattacharyya, «Dynamic Congestion Analysis for Better Traffic Management Using Social Media», em *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2*, vol. 51, S. C. Satapathy e S. Das, Eds. Cham: Springer International Publishing, 2016, pp. 85–95.
- [15] B. Akilesh, N. Kumar, B. Reddy, e M. Singh, «TRAFAN: Road traffic analysis using social media web pages», em *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, Bengaluru, Jan. 2018, pp. 655–659.
- [16] J. Cui, R. Fu, C. Dong, e Z. Zhang, «Extraction of traffic information from social media interactions: Methods and experiments», em *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, Out. 2014, pp. 1549–1554.
- [17] B. Pan, Y. Zheng, D. Wilkie, e C. Shahabi, «Crowd sensing of traffic anomalies based on human mobility and social media», em *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL '13*, Orlando, Florida, 2013, pp. 344–353.
- [18] G. Singh, D. Bansal, S. Sofat, e N. Aggarwal, «Smart patrolling: An efficient road surface monitoring using smartphone sensors and crowdsourcing», *Pervasive Mob. Comput.*, vol. 40, pp. 71–88, Set. 2017.
- [19] A. Allouch, A. Koubaa, T. Abbes, e A. Ammar, «RoadSense: Smartphone Application to Estimate Road Conditions Using Accelerometer and Gyroscope», *IEEE Sens. J.*, vol. 17, n. 13, pp. 4231–4238, Jul. 2017.
- [20] R. Bhoraskar, N. Vankadhara, B. Raman, e P. Kulkarni, «Wolverine: Traffic and road condition estimation using smartphone sensors», em *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, Bangalore, India, Jan. 2012, pp. 1–6.
- [21] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, e H. Balakrishnan, «The pothole patrol: using a mobile sensor network for road surface monitoring», em *Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys '08*, Breckenridge, CO, USA, 2008, p. 29.
- [22] N. Mazhar *et al.*, «Mobile Crowdsensing Application of Road Condition Detection», *Urdu News Headl. Text Classif. Using Differ. Mach. Learn. Algorithms*, p. 25, 2019.
- [23] P. Chen, F. Chen, e Z. Qian, «Road Traffic Congestion Monitoring in Social Media with Hinge-Loss Markov Random Fields», em *2014 IEEE*

International Conference on Data Mining, Shenzhen, China, Dez. 2014, pp. 80–89.

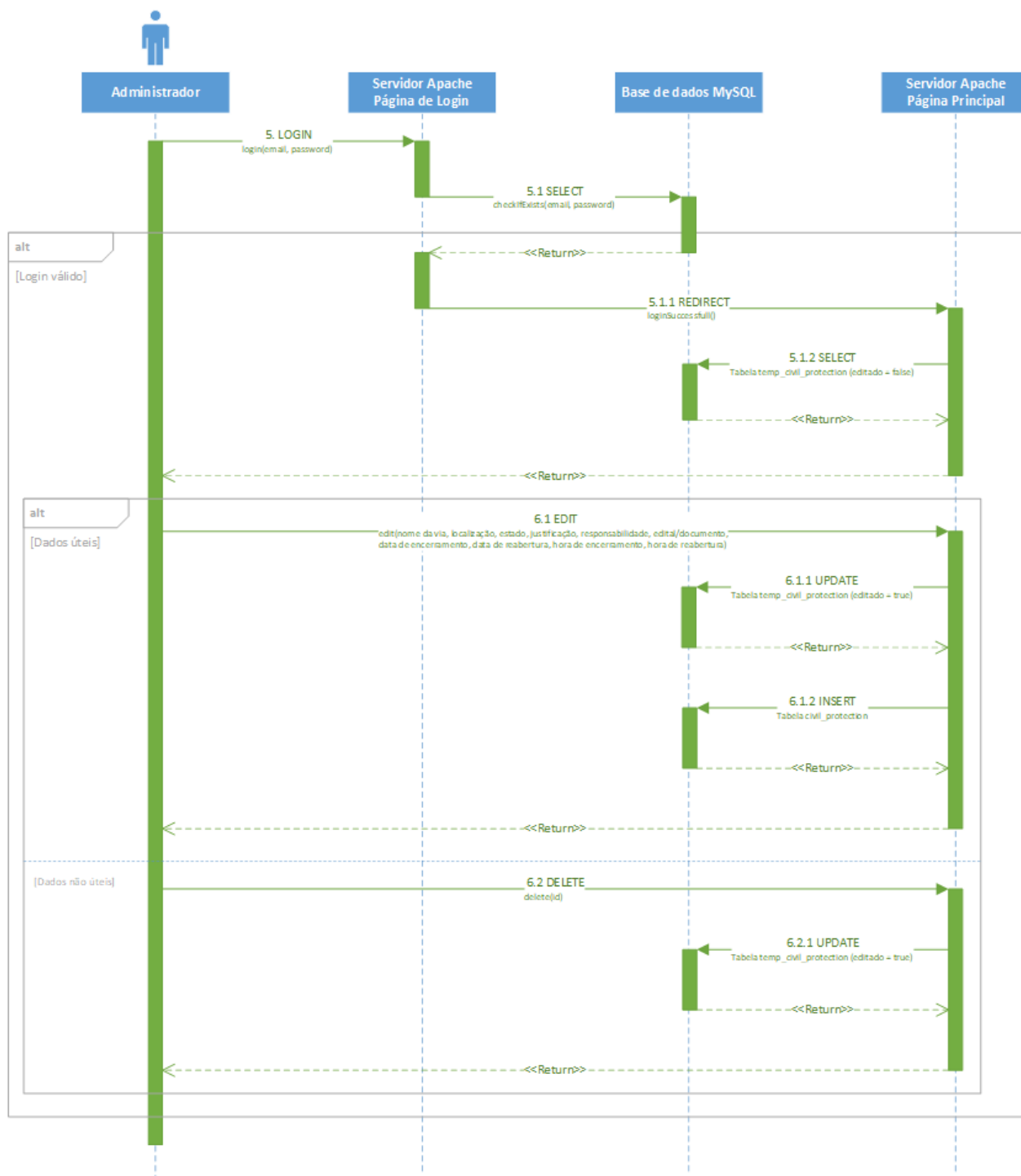
- [24] M. Havlíková, «Likert scale versus Q-table measures – a comparison of host community perceptions of a film festival», *Scand. J. Hosp. Tour.*, vol. 16, n. 2, pp. 196–207, Abr. 2016.

Anexos

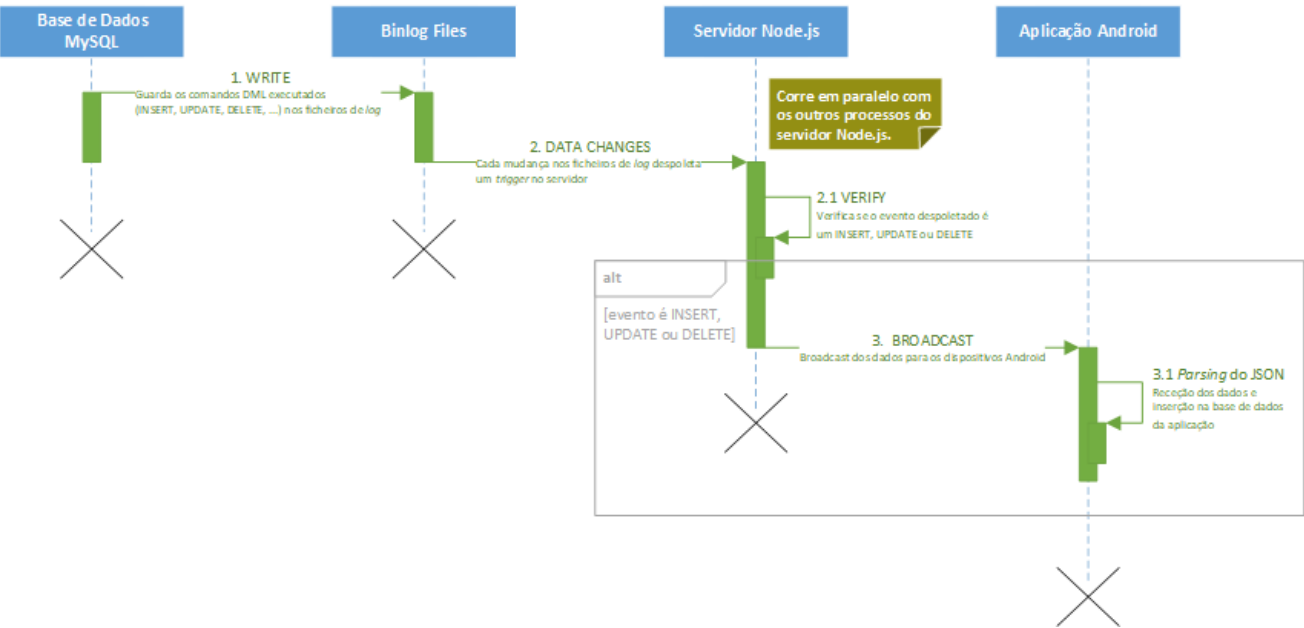
Anexo A- Tratamento dos Dados obtidos a partir do Serviço Regional de Proteção Civil da Madeira



Anexo B – Normalização dos Dados na Página de Administração



Anexo C – Envio dos Dados para a Aplicação



Anexo D – Formulário dos Questionários

Aplicação Móvel para Reduzir o Stress Diário

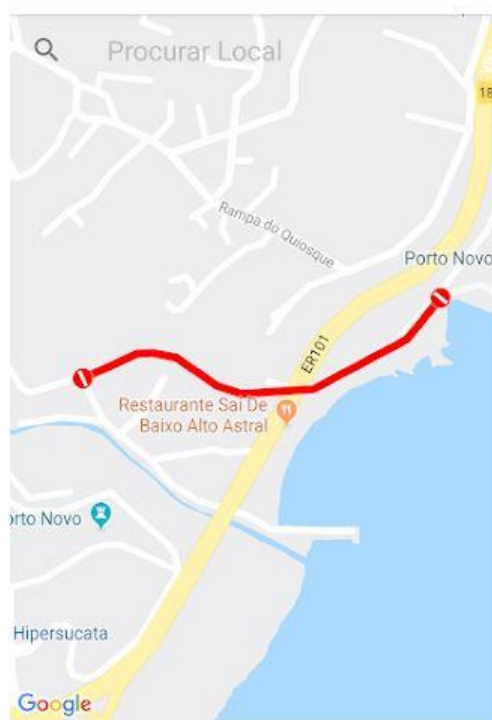
Este questionário tem o intuito de avaliar uma aplicação para Android cujo principal objetivo é o fornecimento de dados relacionados com o tráfego automóvel aos utilizadores.

Para a realização deste questionário é necessário que efetue previamente alguns testes à aplicação fornecida.

*Obrigatório

1. Como avalia a apresentação das estradas encerradas sobre o mapa? *

Marcar apenas um círculo.



1 2 3 4 5

Muito Fraco

☐ ☐ ☐ ☐ ☐

Muito Bom

2. Qual o grau de dificuldade de interpretação dos traços/ícones utilizados na representação das estradas encerradas? *

Marcar apenas um círculo.

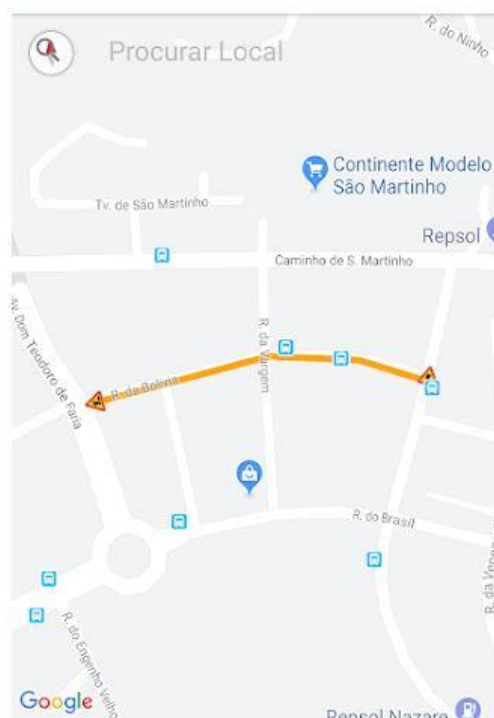
	1	2	3	4	5	
Difícil interpretação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fácil interpretação

3. Existe algum aspeto em particular que deva ser alterado, de modo a melhorar a apresentação das estradas encerradas?

A sua resposta

4. Como avalia a apresentação das estradas condicionadas sobre o mapa? *

Marcar apenas um círculo.



	1	2	3	4	5	
Muito Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito Bom

5. Qual o grau de dificuldade de interpretação dos traços/ícones utilizados na representação das estradas condicionadas? *

Marcar apenas um círculo.

	1	2	3	4	5	
Difícil interpretação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fácil interpretação

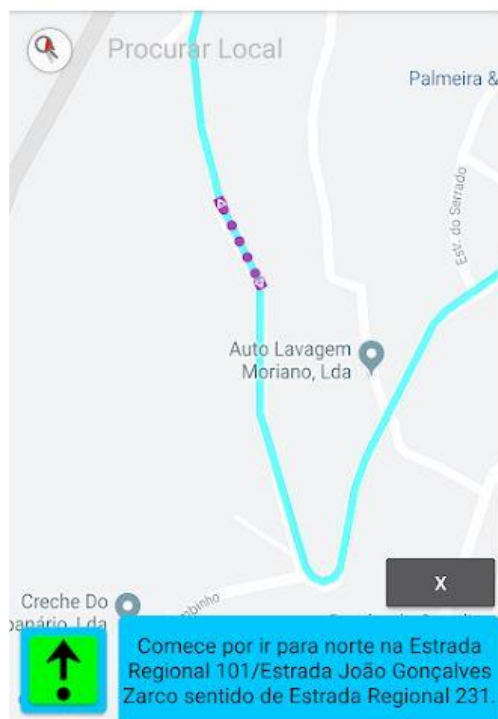
6. Existe algum aspeto em particular que deva ser alterado, de modo a melhorar a apresentação das estradas condicionadas?

A sua resposta

Estradas estreitas e/ou inclinadas

7. Como avalia a apresentação da inclinação das estradas sobre o mapa? *

Marcar apenas um círculo.



	1	2	3	4	5	
Muito Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito Bom

8. Qual o grau de dificuldade de interpretação dos traços/ícones utilizados na representação das estradas com uma inclinação elevada? *

Marcar apenas um círculo.

	1	2	3	4	5	
Difícil interpretação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fácil interpretação

9. Existe algum aspeto em particular que deva ser alterado, de modo a melhorar a apresentação das estradas com uma inclinação elevada?

A sua resposta

10. Como avalia a apresentação da largura das estradas sobre o mapa (estradas estreitas)? *

Marcar apenas um círculo.



	1	2	3	4	5	
Muito Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito Bom

11. Qual o grau de dificuldade de interpretação dos traços/ícones utilizados na representação das estradas estreitas? *

Marcar apenas um círculo.

1 2 3 4 5

Difícil interpretação

☐☐☐☐☐

Fácil interpretação

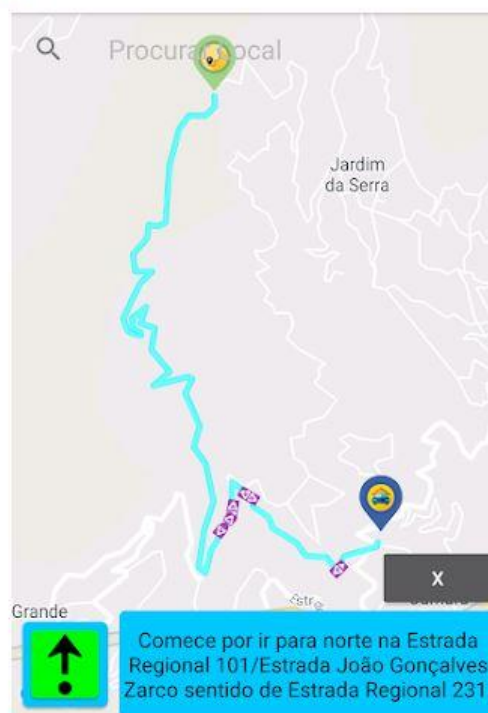
12. Existe algum aspeto em particular que deva ser alterado, de modo a melhorar a apresentação das estradas estreitas?

A sua resposta

Navegação GPS (turn-by-turn)

13. Como avalia a navegação GPS (turn-by-turn)? *

Marcar apenas um círculo.



1 2 3 4 5

Muito Fraco

☐☐☐☐☐

Muito Bom

14. Qual o grau de dificuldade de interpretação dos ícones de início e de destino? *

Marcar apenas um círculo.

	1	2	3	4	5	
Difícil interpretação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fácil interpretação

15. Avalie os seguintes aspetos relacionados com a navegação GPS (turn-by-turn). *

Marcar apenas um círculo.

	Muito Fraco	Fraco	Suficiente	Bom	Muito Bom
Apresentação da rota no mapa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Instruções escritas da próxima rota a seguir	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Instruções faladas da próxima rota a seguir	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apresentação por setas da próxima rota a seguir	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

16. Existe algum aspeto em particular que deva ser alterado, de modo a melhorar a navegação GPS (turn-by-turn)?

A sua resposta

Dados obtidos a partir do grupo do Facebook "Ocorrências Madeira"

17. Como avalia as informações obtidas a partir do grupo "Ocorrências Madeira"? *

Marcar apenas um círculo.



1 2 3 4 5

Muito Fraco

☐☐☐☐☐

Muito Bom

18. Qual o grau de importância das informações obtidas a partir do grupo "Ocorrências Madeira"? *

Marcar apenas um círculo.

1 2 3 4 5

Irrelevante

☐☐☐☐☐

Imprescindível

19. Como avalia a apresentação no mapa dos acidentes postados no grupo "Ocorrências Madeira"? *

Marcar apenas um círculo.



1 2 3 4 5

Muito Fraco

☐☐☐☐☐

Muito Bom

20. Existe algum aspeto em particular que deva ser alterado, de modo a melhorar a apresentação dos dados obtidos a partir do grupo "Ocorrências Madeira"?

A sua resposta

Sugestões

21. Existe algum aspeto em particular que deva ser alterado, de modo a melhorar o funcionamento da aplicação (E.g. forma como são apresentados os dados, adicionar informações extra...) ?

A sua resposta

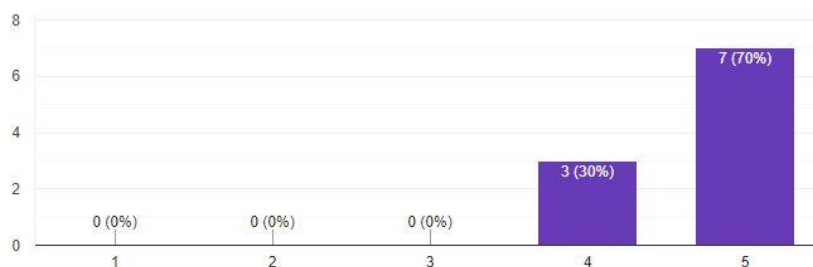
22. Que funcionalidade(s) acrescentaria à aplicação?

A sua resposta

Anexo E – Respostas aos Questionários

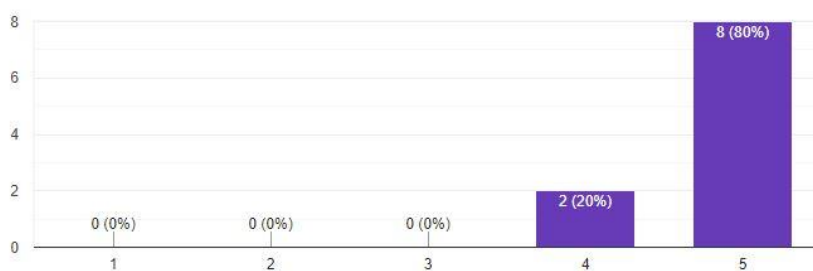
1. Como avalia a apresentação das estradas encerradas sobre o mapa?

10 respostas



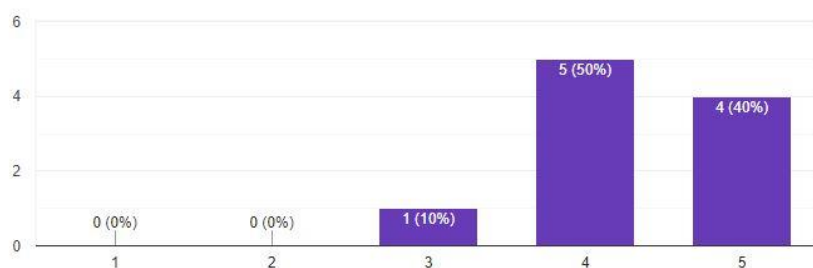
2. Qual o grau de dificuldade de interpretação dos traços/ícones utilizados na representação das estradas encerradas?

10 respostas



4. Como avalia a apresentação das estradas condicionadas sobre o mapa?

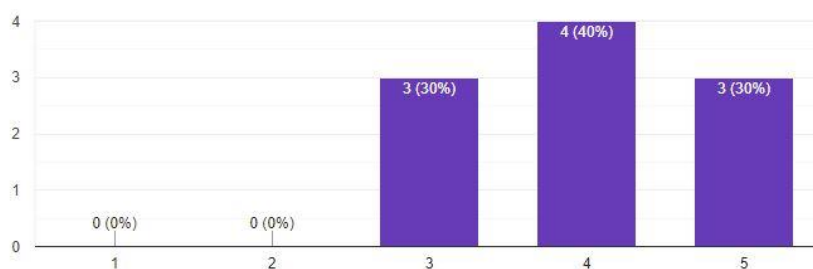
10 respostas



5. Qual o grau de dificuldade de interpretação dos traços/ícones utilizados na representação das estradas condicionadas?



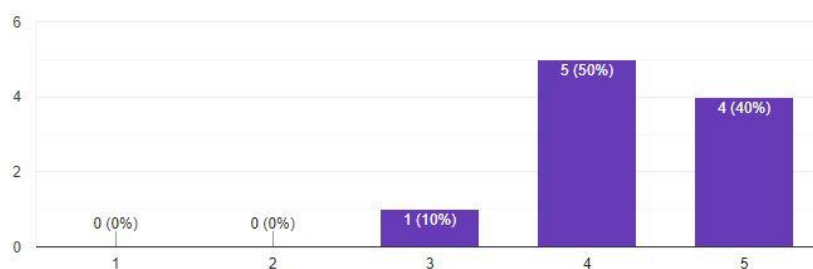
10 respostas



7. Como avalia a apresentação da inclinação das estradas sobre o mapa?



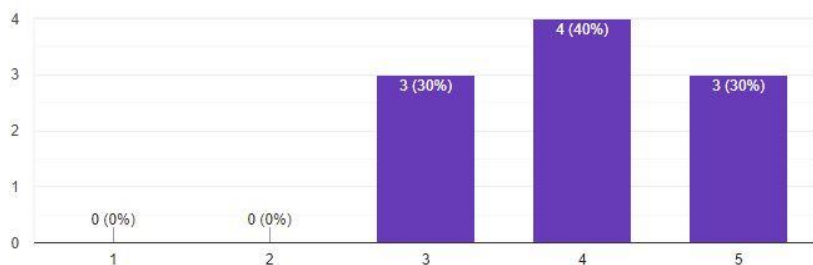
10 respostas



8. Qual o grau de dificuldade de interpretação dos traços/ícones utilizados na representação das estradas com uma inclinação elevada?



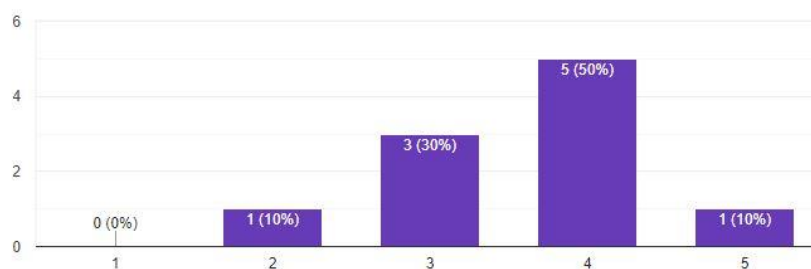
10 respostas



10. Como avalia a apresentação da largura das estradas sobre o mapa (estradas estreitas)?



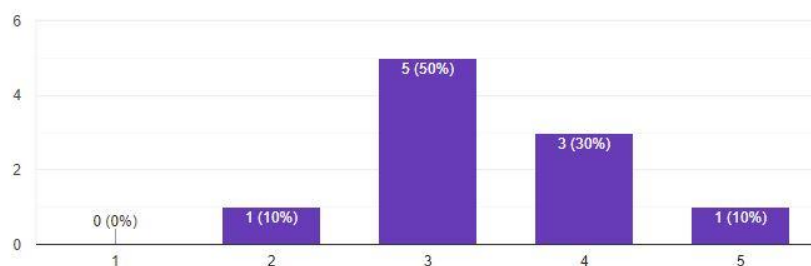
10 respostas



11. Qual o grau de dificuldade de interpretação dos traços/ícones utilizados na representação das estradas estreitas?



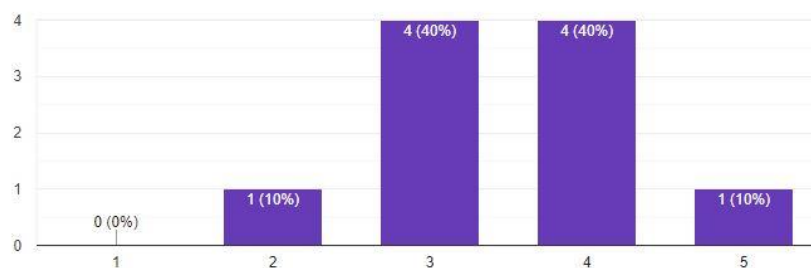
10 respostas



13. Como avalia a navegação GPS (turn-by-turn)?

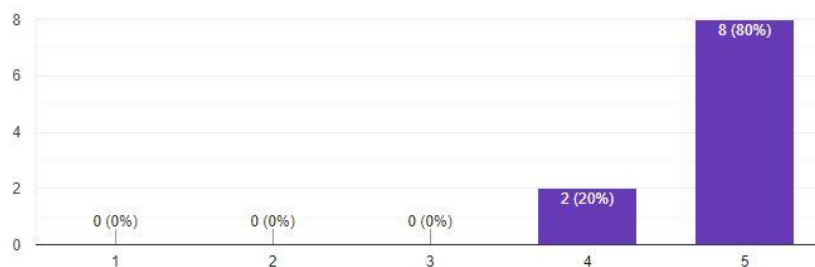



10 respostas

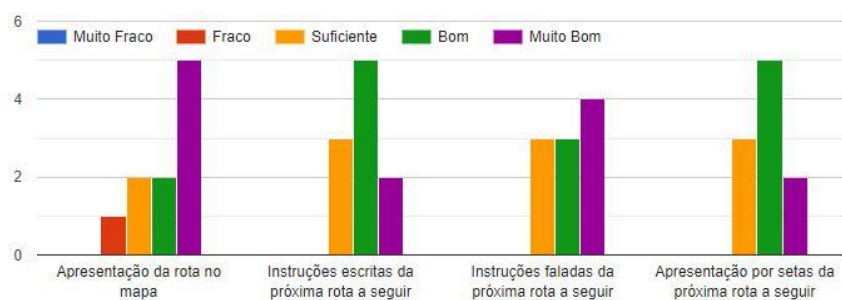



14. Qual o grau de dificuldade de interpretação dos ícones de início e de destino? 

10 respostas

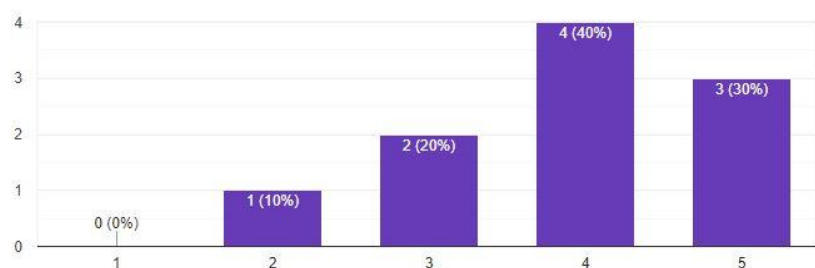


15. Avalie os seguintes aspetos relacionados com a navegação GPS (turn-by-turn). 



17. Como avalia as informações obtidas a partir do grupo "Ocorrências Madeira"? 

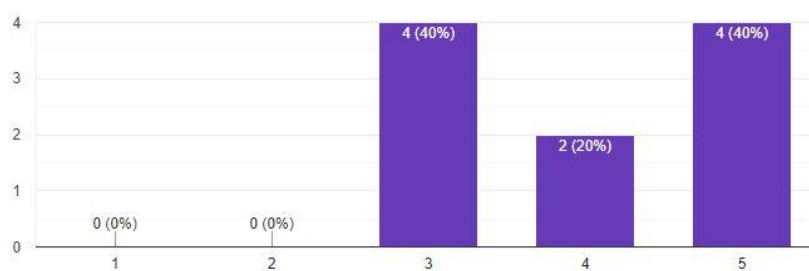
10 respostas



18. Qual o grau de importância das informações obtidas a partir do grupo "Ocorrências Madeira"?



10 respostas



19. Como avalia a apresentação no mapa dos acidentes postados no grupo "Ocorrências Madeira"?

10 respostas

